J Syst Sci Syst Eng
DOI: 10.1007/s11518-015-5266-0
ISSN: 1004-3756 (Paper) 1861-9576 (Online)
CN11-2983/N

SCHEDULING IN THE SERVICE INDUSTRIES: AN OVERVIEW

Michael Pinedo[1] Christos Zacharias[2] Ning Zhu[3]

[1]*Stern Business School, New York University, New York, NY 10012*

mpinedo@stern.nyu.edu (✉)

[2]*School of Business Administration, University of Miami, Coral Gables, FL 33146*

czacharias@miami.edu

[3]*College of Economics and Management, Tianjin University, Tianjin, China*

zhuning@tju.edu.cn

Abstract

Scheduling plays an important role in many different service industries. In this paper we provide an overview of some of the more important scheduling problems that appear in the various service industries. We focus on the formulations of such problems as well as on the techniques used for solving those problems. We consider five areas of scheduling in service industries, namely (i) project scheduling, (ii) workforce scheduling, (iii) timetabling, reservations, and appointments, (iv) transportation scheduling, and (v) scheduling in entertainment. The first two areas are fairly general and have applications in many different service industries. The third, fourth and fifth areas are more related to some very specific service industries, namely the hospitality and health care industries, the transportation industries (of passengers as well as of cargo), and the entertainment industries. In our conclusion section we discuss the similarities and the differences between the problem formulations and solution techniques used in the various different industries and we also discuss the design of the decision support systems that have been developed for scheduling in the service industries.

1. Introduction

Scheduling applications in the service industries are ubiquitous. Due to the inherent non-stationarity of service businesses, scheduling is a very important aspect of management in a variety of service industries, including health care, hospitality, transportation, and entertainment industries. This paper presents a tutorial of some of the major scheduling application areas in the service industries. It focuses, in particular, on three aspects of scheduling in the service industries, namely (i) the most basic scheduling paradigms relevant to the service industries, (ii) optimization techniques and/or heuristics that are used in practice, and (iii) examples of specific real world applications.

One can make a distinction between static scheduling and dynamic scheduling. In static scheduling applications, one would not expect a schedule to change much over time; a schedule is typically cast in stone. A typical example of such schedule is a quarterly flight schedule of an airline. Since such a static schedule is not expected to undergo many changes, the amount of computer time allocated for finding an optimal solution may be substantial; such a schedule typically does not have to be generated in real time. In dynamic scheduling applications, on the other hand, one would expect a schedule to change frequently. When schedules have to be generated and regenerated regularly, the optimization may have to be done in real time. Because of the many changes that are then expected, one important characteristic of a schedule is its robustness. An example of a dynamic scheduling application can be a resource constrained project at a consulting firm. In practice, dynamic scheduling is often done manually, rather than through a decision support system.

It has been the experience that the analysis of a dynamic scheduling problem is often harder than the analysis of its corresponding static scheduling counterpart. However, in the development of a procedure for a dynamic scheduling problem it is always helpful to know which procedure(s) are most appropriate for the scheduling of its static counterpart.

In this paper we provide an overview of five different scheduling areas in service industries. The first two areas are fairly general and have applications in many different service industries, namely

(i) project scheduling, and

(ii) workforce scheduling.

The third, fourth and fifth areas focus on specific service industries, namely

(iii) hospitality and health care industries,

(iv) transportation industries, and

(v) professional sports and entertainment.

The first area, project scheduling, has many applications in management consulting, accounting and auditing, as well as in systems implementations. The second area, workforce scheduling, consists of two parts, one being shift scheduling (important in call centers) and the other being crew scheduling (important in transportation). The third area considers timetabling, reservations, and appointments. The corresponding section consists of three subsections. The first subsection goes into the basics of timetabling, which has many applications in the hospitality industries as well as in the field of education. The second subsection covers interval scheduling and reservation systems modeling, which are closely related to timetabling. The last subsection discusses a more special case of timetabling, namely appointment scheduling, which is very important in health care. The fourth area deals with transportation scheduling. The corresponding section consists of four subsections. The first one focuses on urban transit scheduling, the second one on maritime scheduling, the third one on aviation scheduling and the last one on emergency operations scheduling. The fifth and last area covers scheduling in professional sports and entertainment. This section consists of two subsections. The first one focuses on tournament scheduling in professional sports and the second one on network broadcast scheduling. In the very last section, we present our conclusions, discuss the design and development of decision support systems, and make suggestions for future research.

There are other scheduling areas in service industries that are not covered in this tutorial. However, the areas that are covered seem to be very representative of common scheduling problems in service industries. The goal of this paper is to provide a rudimentary overview of scheduling applications and the methods and techniques being used in the service industries. Since it is more a tutorial rather than a detailed survey of the entire literature, the reference list is not exhaustive. There are in the literature many more papers on each topic we discuss.

2. Project Scheduling

Examples of project scheduling are ubiquitous in the service industries; they include consulting projects, systems installation projects, maintenance and repair projects, and so on. Consulting projects may include also the annual auditing processes that must be done at every public company by independent accounting (CPA) firms. A systems installation project may involve the installation of a large computer system at a corporation or the implementation of a large ERP system; these types of projects can take several years. A maintenance and repair project may be the annual overhaul of a major manufacturing or power generation facility; such a facility may be forced to stop its production in order for the maintenance to take place.

Project scheduling in service industries tend to be intrinsically very different from project scheduling in manufacturing industries. Installing a large ERP system at a major company does not have much in common with the building of an aircraft carrier or a nuclear submarine.

In this section we discuss the basics of project scheduling. The first subsection focuses mainly on the precedence constraints imposed on the activities and the resulting critical paths. The second subsection considers more general project scheduling problems that have, in addition to the precedence constraints, resource constraints.

2.1 Precedence Constraints and Critical Paths

A generic project scheduling problem can be described as follows: consider the scheduling of a number of jobs or activities that are subject to precedence constraints. A job or activity can start only when all its predecessors have been completed. The objective is to minimize the total project completion time while adhering to the precedence constraints. Such problem is considered a standard project scheduling problem.

Research in project scheduling started in the 1950s. These efforts resulted in the classical technique usually referred to as the Critical Path Method (CPM). There is an extensive literature, spanning decades, in the field of project scheduling and in the Critical Path Method; see, for example, Walker and Sayer (1959), Moder and Philips (1970), Wiest and Levy (1977), and Demeulemeester and Herroelen (2002).

Since, especially in service industries, activity durations are often random, a fair amount of effort has been put into the development of critical path techniques for random durations; one such technique is known as the Project Evaluation and Review Technique (PERT), see the Department of the Navy Report "PERT" (1958), Fulkerson (1962), Elmaghraby (1967), and Sasieni (1986).

Another version of the project scheduling problem assumes that the durations of activities can be determined in advance by the project manager. A project manager may have some control over the durations of different activities by allocating selectively more resources (e.g., people) to some activities. A project may have a deadline and a completion after the deadline may entail a penalty; the project manager, therefore, has to analyze the trade-off between the penalties incurred by completing the project late and the additional costs incurred by shortening the durations of selected activities by allocating more resources, see Talbot (1982). This process is in the literature typically referred to as crashing.

2.2 Project Scheduling with Resource Constraints

Another more general version of the basic project scheduling problem assumes that a job's processing requires additional resources of different types, say some special equipment or specific experts. Consider, for example, a workforce that consists of various different pools of people with each pool having of a fixed number of people with a specific skill set. Because of the pools' limitations, it may sometimes occur that two jobs cannot be processed at the same time, even though both are allowed to start as far as the precedence constraints are concerned. The total number of people the two jobs require from a given pool may be larger than the number available in that pool, making it impossible to process two jobs at the same time. This type of problem is typically referred to as project scheduling with resource constraints. Resource constraints typically make project scheduling problems considerably harder. A significant amount of research in the past has focused on project scheduling subject to resource constraints, see Patterson (1984), Blazewicz et al. (1986), Kolish (1995), Brucker et al. (1999), and Neumann et al. (2001).

The basic project scheduling problem with precedence constraints but without any resource constraints is very easy from a computational point of view. Optimal solutions can be found with very little computational effort. However, project scheduling problems with resource constraints are typically strongly NP-Hard.

A project scheduling problem subject to resource constraints typically can be formulated as a Mixed Integer Program (MIP). In order to formulate this problem as an integer program, assume that all processing times are fixed and integer. Let W_ℓ denote the total number of people available in pool ℓ and let $W_{\ell j}$ denote the number of people job j requires from pool ℓ for its processing. Let A denote the set of precedence constraints. Introduce a dummy job $n+1$ with zero processing time. Job $n+1$ succeeds all other jobs, i.e., all jobs without successors have an arc emanating to job $n+1$. Let x_{jt} denote a $0-1$ variable that assumes the value 1 if job j is completed exactly at time t and the value 0 otherwise. So the number of operators job j needs from pool ℓ in the interval $[t-1,t]$ is

$$W_{\ell j} \sum_{u=t}^{t+p_j-1} x_{ju} .$$

Let H denote an upper bound on the makespan. A simple, but not very tight, bound can be obtained by setting

$$H = \sum_{j=1}^{n} p_j \ .$$

So the completion time of job j can be expressed as

$$\sum_{t=1}^{H} t x_{jt} \ .$$

And the makespan as

$$\sum_{t=1}^{H} t x_{n+1,t} \ .$$

An integer program can now be formulated as follows:

Minimize $\sum_{t=1}^{H} t x_{n+1,t}$

subject to

$$\sum_{t=1}^{H} t x_{jt} + p_k - \sum_{t=1}^{H} t x_{kt} \le 0, \quad \text{for } j \to k \in A$$

$$\sum_{j=1}^{n} \left(W_{\ell j} \sum_{u=t}^{t+p_j-1} x_{ju} \right) \le W_\ell \ , \quad \text{for} \begin{pmatrix} \ell = 1,\ldots,N_p; \\ t = 1,\ldots,H \end{pmatrix}$$

$$\sum_{t=1}^{H} x_{jt} = 1, \text{ for } j = 1,\ldots,n \ .$$

The objective of the integer program is to minimize the makespan. The first set of constraints ensures that the precedence constraints are enforced, i.e., if job j is followed by job k, then the completion time of job k has to be greater than or equal to the completion of job j plus p_k. The second set of constraints ensures that the total demand for pool ℓ at time t does not surpass the availability of pool ℓ. The third set of constraints ensures that each job is processed.

Since this integer program is very hard to solve when the number of jobs is large and the time horizon is long, it is typically tackled with heuristics. It turns out that even special cases of this problem are quite hard. However, for a number of important special cases heuristics have been developed that have been proven to be quite effective.

Over the last decade research in project scheduling has started to focus on various types of resource constraints. Traditional resources may be referred to as “renewable”, since they will always be available. A renewable resource could be a person, i.e., a specific expert, who is on the payroll of a company. After this person has lended his hand in the completion of one activity and the activity has been completed, he could be assigned to another activity. However, other resources may be referred to as “nonrenewable”. Such resources would actually be consumed and a certain supply was available at the outset. Of late, research has started to focus on project scheduling with renewable as well as nonrenewable resources. A specific example of a nonrenewable resource is working capital. Such nonrenewable resource constraints are basically equivalent to budgetary constraints.

In particular, project scheduling subject to resource constraints with random activity durations has not received much research attention in the past. This particular area seems to be in need of new research ideas.

3. Workforce Scheduling

Workforce scheduling is a very important aspect of many service industries, since schedules have to be created in such a way that they will be able to deal with fluctuating and random demand. The application areas include nurse scheduling in hospitals, operator scheduling in call centers, and so on. Clearly, an enormous amount of research has been done on personnel scheduling, resulting in a host of survey papers and books; see, for example, Tien and Kamiyama (1982), Burgess and Busby (1992), Nanda and Browne (1992), and Burke et al. (2004). This section consists of two subsections: the first one deals with shift scheduling, which is very important in call centers, and the second one deals with crew scheduling, which is very important in transportation industries.

3.1 Shift Scheduling

In this subsection we consider personnel scheduling problems with cycles that are fixed in advance. In certain settings the cycle may be a single day, while in others it may be a week or a number of weeks. Each work assignment pattern within a cycle has its own cost and the objective is to minimize the total cost.

The problem can be formulated as follows: A predetermined cycle consists of m time intervals or periods. The lengths of the periods do not necessarily have to be identical. During period i, $i=1,\ldots,m$, the presence of b_i personnel is required. The number b_i is, of course, an integer. There are n different shift patterns and each employee is assigned to one and only one pattern. Shift pattern j is defined by a vector $(a_{1j}, a_{2j}, \ldots, a_{mj})$. The value a_{ij} is either 0 or 1; it is a 1 if period i is a work period and 0 otherwise. Let c_j denote the cost of assigning a person to shift j and x_j the (integer) decision variable representing the number of people assigned to shift j. The problem of minimizing the total cost of assigning personnel to meet demand can be formulated as the following integer programming problem:

Minimize $c_1 x_1 + c_2 x_2 + \ldots + c_n x_n$

subject to

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \geq b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \geq b_2, \\ \qquad\qquad\vdots \qquad\qquad\qquad \text{for } j = 1,\ldots,n, \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \geq b_m, \\ \qquad\qquad x_j \geq 0. \end{cases}$$

with $x_1,\ldots,x_n$ integer. In matrix form this integer program is written as follows.

Minimize $\bar{c}\,\bar{x}$

subject to

$$\mathbf{A}\bar{x} \geq \bar{b}\ .$$

Such an integer programming problem is known to be strongly NP-hard in its most general form. However, the $\mathbf{A}$ matrix may often exhibit a special structure. For example, shift j, $(a_{1j},\ldots a_{mj})$, may contain a *contiguous* set of 1's (a contiguous set of 1's implies that there are no 0's in between 1's). However, the *number* of 1's may often vary from shift to shift, since it is possible that some shifts have to work longer hours or more days than other shifts.

Even though the integer programming formulation of the general personnel scheduling problem (with an arbitrary 0 − 1 **A** matrix) is NP-hard, the special case with each column containing a contiguous set of 1's is easy. It can be shown that the solution of the linear programming relaxation is always integer. There are several other important special cases that are solvable in polynomial time. Many papers have focused on a number of special cases of the problem described above; see, for example, Bartholdi et al. (1980), Burns and Carter (1985), Burns and Koop (1987), Emmons (1985), Emmons and Burns (1991), Gawande (1996), Hung and Emmons (1993).

3.2 Crew Scheduling

Another type of workforce scheduling involves crew scheduling, which is a form of workforce scheduling that is very important in transportation industries, e.g., aviation and trucking. Crew scheduling is from a mathematical point of view very different from shift scheduling. It has also received a lot of research attention; see, for example, Bodin et al. (1983), Marsten and Shepardson (1981), Stojkovich et al. (1998).

Crew scheduling problems are very important in the transportation industry, especially in the airline industry. The underlying model is different from the models considered in the previous sections and so are the solution techniques.

Consider a set of m jobs, e.g., flight legs. A flight leg is characterized by a point of departure and a point of arrival, as well as an approximate time interval during which the flight has to take place. There is a set of n feasible and permissible combinations of flight legs that one crew can handle, e.g., round trips or tours (the number n usually is very large). A round trip may consist of several flight legs, i.e., a plane may leave city A for city B, then go to city C, before returning to city A. Any given flight leg may be part of many round trips. Round trip j, $j = 1, \ldots, n$, has a cost c_j. Setting up a crew schedule is equivalent to determining which round trips should be selected and which ones not. The objective is to choose a set of round trips with a minimum total cost in such a way that each flight leg is covered exactly once by one and only one round trip.

In order to formulate this crew scheduling problem as an integer program some notation is required. If flight leg i is part of round trip j, then a_{ij} is 1, otherwise a_{ij} is 0. Let x_j denote a 0 − 1 decision variable that takes the value 1 if round trip j is selected and 0 otherwise. The crew scheduling problem can be formulated as the following integer program.

$$\text{Minimize } c_1 x_1 + c_2 x_2 + \ldots + c_n x_n$$
subject to
$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = 1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = 1, \\ \qquad\qquad\qquad\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = 1. \\ x_j \in \{0,1\} \qquad \text{for } j = 1, \ldots, n. \end{cases}$$

Each column in the **A** matrix is a round trip and each row is a flight leg that must be covered exactly once by one round trip. The optimization problem is then to select, at minimum cost, a set of round trips that satisfies the constraints. The constraints in this problem are often called the partitioning equations and this integer programming problem is referred to as the *Set Partitioning* problem (see Appendix A). For a feasible solution $(x_1, \ldots, x_n)$, the variables that are equal to 1 are referred to as the partition. In what follows we denote a partition l by $J^l = \{j \mid x_j^l = 1\}$.

This problem is known to be NP-hard. Many heuristics as well as enumeration schemes (branch-and-bound) have been proposed for this problem. In many of these approaches the concept of *row prices* is used. The vector $\overline{\rho}^l = (\rho_1^l, \rho_2^l, \ldots, \rho_m^l)$ is a set of feasible row prices corresponding to partition J^l satisfying

$$\sum_{i=1}^{m} \rho_i^l a_{ij} = c_j \quad j \in J^l \ .$$

The price ρ_i^l may be interpreted as an estimate of the cost of covering job (flight leg) i using solution J^l. There are usually many feasible price vectors for any given partition.

The row prices are of crucial importance in computing the change in the value of the objective if a partition J^1 is changed into partition J^2. If $Z^1(Z^2)$ denotes the value of the objective corresponding to partition 1 (2), then

$$Z^2 = Z^1 - \sum_{j \in J^2} \left(\sum_{i=1}^{m} \rho_i^1 a_{ij} - c_j \right).$$

The quantity

$$\sigma_j = \sum_{i=1}^{m} \rho_i^1 a_{ij} - c_j$$

can be interpreted as the potential savings with respect to the first partition to be obtained by including column j. It can be shown that if

$$\sum_{i=1}^{m} \rho_i^1 a_{ij} \le c_j \quad j = 1, \ldots, n \,,$$

for any set of feasible row prices $\overline{\rho}^1$ corresponding to partition J^1, then solution J^1 is optimal.

Based on the concept of row prices the following simple heuristic can be used for finding better solutions, given a partition J^1 and a corresponding set of feasible row prices $\overline{\rho}^1$. The goal is to find a better partition J^2. In the heuristic the set N denotes the indices of the columns that are candidates for inclusion in J^2.

Algorithm 3.2.1: Column Selection in Set Partitioning

Step 1

Set $J^2 = \varnothing$ and $N = \{1, 2, \ldots, n\}$.

Step 2

Compute the potential savings

$$\sigma_j = \sum_{i=1}^{m} \rho_i^1 a_{ij} - c_j \quad j = 1, \ldots, n$$

Find the column k in N with the largest potential savings

$$\sum_{i=1}^{m} \rho_i^1 a_{ik} - c_k \ .$$

Step 3

For $i = 1, \ldots m$, if $a_{ik} = 1$ set $a_{ij} = 0$ for all $j \ne k$.

Step 4

Let $J^2 = J^2 \bigcup \{k\}$ and $N = N - \{k\}$.

Delete from N all j for which $a_{ij} = 0$ for all $i = 1, \ldots, m$.

Step 5

If $N = \varnothing$ STOP, otherwise go to Step 2.

When the problem becomes very large, it is necessary to adopt more sophisticated approaches, namely Branch-and-Bound, Branch-and-Price, and Branch-Cut-and-Price. The bounding techniques in Branch-and-Bound are often based on a technique called Lagrangean Relaxation. Branch-Cut-and-Price combines branching with so-called cutting planes techniques and has been used to solve real world problems arising in the airlines industry with considerable success.

4. Timetabling, Reservations, and Appointments

In the hospitality industries, education, and health care there are many timetabling, reservation, and appointment scheduling problems. These problems often tend to be mathematically related to one another and may require similar solution techniques, which include integer programming formulations as well as graph theoretic approaches.

4.1 Timetabling

Timetabling refers to a class of generic scheduling problems with numerous applications in education, transportation, health care, and other service industries. The applications described in this section are related to some of the applications described in subsequent sections.

In the most basic timetabling model there are typically n activities or jobs to be scheduled. In a timetabling problem an activity (say, for example, a meeting) can only be scheduled if a given set of very specific people and/or resources are all available at the time. So an activity can be scheduled at any time as long as all the necessary people and/or resources are available in the time interval selected. The availability of the people may be subject to constraints and the constraints may imply that certain subsets of activities cannot be done at the same time, because a particular person cannot participate in two different activities at the same time. A typical objective of the scheduling problem may be to finish all the activities (e.g., meetings) in the shortest possible time, i.e., to minimize the makespan. In other words, to finish the last activity as early as possible. In a more general timetabling problem the timing of activity j may also be constrained by an earliest starting time r_j and a latest completion time d_j.

A distinction can be made between several different types of timetabling problems: One type of timetabling problem assumes that all people involved have the same skill set and are therefore interchangeable, i.e., they represent a homogeneous workforce. The total number of people in the workforce is W and in order to do activity j W_j operators have to be present. If the sum of the people required by activities j and k is larger than W (i.e., $W_j + W_k > W$), then activities j and k may not overlap in time. (Such a constraint would be equivalent to a (renewable) resource constraint as described in the previous section.) This type of timetabling problem may be referred to as timetabling with workforce or personnel constraints.

In a second type of timetabling problem each person (or resource) has its own identity or skill set, i.e., they represent a heterogenous workforce. Each activity or job now requires a specific subset of the people. In order for an activity to be scheduled all the people in its subset have to be available. Two activities that need the same person cannot be done at the same time. This type of timetabling problem is in what follows referred to as timetabling subject to operator constraints.

This second type of timetabling can occur in many different settings. Consider, for example, a large repair shop for aircraft engines. In order to do a certain type of repair it is necessary to have a certain type of person and a certain type of tool available at the same time. Since a given type of person may be required for a certain type of repair, timetabling may become necessary. A second example of this type of timetabling occurs when meetings have to be scheduled. Each meeting requires a given set of people to attend and each meeting has to be assigned to a time period in which all who have to attend are available. The meeting rooms also correspond to resources. A third example of this type of timetabling occurs when exams have to be scheduled. Each person represents a student (or a group of students) and two exams that have to be taken by the same student (or groups of students) cannot be scheduled at the same time. The objective is to schedule all the exams within a given time period, say one week. It is therefore necessary to minimize the makespan.

It turns out that timetabling problems are very closely related to graph coloring problems. Consider a timetabling problem with operators, each having his own identity and skill set (an operator may also be equivalent to a specific piece of machinery, a fixture, or a tool). A given activity either needs or does not need any specific operator or tool. Each activity needs for its execution a specific subset of the operators and/or tools. If two activities require the same operator, then they cannot be done at the same time.

In a feasibility version of this problem, the goal is to find a schedule or timetable that completes all n activities within a given time horizon H. In the optimization version, the objective is to do all the activities and minimize the makespan.

Even the special case with all activity durations being equal does not allow for an easy solution. Consider first the feasibility version with all durations being equal to 1. Finding for this case a conflict-free timetable is structurally equivalent to a very well-known node coloring problem in graph theory. In this node coloring problem a graph is constructed by representing each activity as a node. Two nodes are connected by an arc if the two activities require the same operator(s). The two activities, therefore, cannot be scheduled in the same time slot. If the length of the time horizon is H time slots, then the question boils down to the following: can the nodes in the graph be colored with H different colors in such a way that no two connected nodes receive the same color? This is clearly a feasibility problem. The associated optimization problem is to determine the minimum number of colors needed to color the nodes of the graph in such a way that no two connected nodes have the same color. This minimum number of colors is in graph theory referred to as the chromatic number of the graph and is equivalent to the makespan in the timetabling problem.

There are a number of heuristics for this timetabling problem with durations equal to 1. In this section we describe only one such procedure, namely the one that is due to Brelaz (1979). First some graph theory terminology is needed. The degree of a node is the number of arcs connected to a node. In a partially colored graph, the saturation level of a node is the number of differently colored nodes already connected to it. In the coloring process, the first color to be used is labeled Color 1, the second Color 2, and so on.

Algorithm 4.1.1: Graph Coloring Heuristic

Step 1

Arrange the nodes in decreasing order of their degree.

Step 2

Color a node of maximal degree with Color 1.

Step 3

Choose an uncolored node with maximal

saturation level.

If there is a tie, choose any one of the nodes with maximal degree in the uncolored subgraph.

Step 4

Color the selected node with the color with the lowest possible number.

Step 5

If all nodes are colored, STOP. Otherwise go to Step 3.

The structure of the heuristic described above is quite typical for this type of optimization problem. It follows the "path of the most resistance". It tries to schedule early on in the scheduling process those parts of the problem that appear to be the hardest to schedule and that maybe subject to the most constraints. The rationale behind such a heuristic is obvious. Early on in the process it may still be possible to schedule those parts of the problem that appear to be hard to schedule. If those parts of the problem are postponed to a later stage of the scheduling process, they may actually end up to be impossible to schedule.

There is a very extensive literature in the field of timetabling. A series of conferences on time tabling has led to a number of proceedings on this topic, see Burke and Ross (1996), Burke and Carter (1998), Burke and Erben (2001), Burke and De Causmaecker (2003), Burke and Trick (2004), Burke and Rudova (2006). For the literature on examination timetabling, see Carter (1986), and Burke et al. (1996).

However, it is clear that there are still many open problems in the timetabling area. First, in this subsection we have only considered two types of workforces: a completely homogeneous workforce (i.e., all individuals are identical) and the completely heterogeneous workforce (each individual has its very own identity and is not interchangeable with anyone else). We have not considered any mixtures or hybrids of the two models described above. Such problems are actually quite common in practice and clearly very hard.

4.2 Interval Scheduling and Reservations

Interval scheduling problems are ubiquitous in reservation systems in the hospitality industries, e.g., hotels, car-rentals, etc. Consider the following reservation model: There are m resources in parallel and n activities. Activity j has a release date r_j, a due date d_j, and a weight w_j. As stated before, all data are integer. The fact that there is no slack between release date and due date implies that

$$p_j = d_j - r_j \ .$$

If we decide to do activity j, then it has to be done within the specified time frame. However, it may be the case that activity j cannot be done by just any one of the m resources; it may have to be done by a resource that belongs to a specific subset of the m resources, namely subset M_j. When all activities have equal weights, the objective is to maximize the number of activities done. In contrast, when the activities have different weights, the objective is to maximize the weighted number of activities scheduled. A weight may be equivalent to a profit that is made by doing the activity. In a more general model the weight of activity j may also depend on the resource to which it is assigned, i.e., the weight is w_{ij} (i.e., the profit depends on the activity as well as on the resource).

Example 4.2.1: *A Car Rental Agency* Consider a car rental agency with four types of cars: subcompact, midsize, full size and sport-utility. Of each type there are a fixed number available. When customer j calls to make a reservation for p_j days, he may, for example, request a car of either one of two types and will accept the price quoted by the agency for either type. The set M_j for such a customer includes all cars belonging to the two types. The profit made by the agency for a car of type i is π_i dollars per day. So, the weight of this particular reservation is $w_{ij} = \pi_i p_j$.

However, if customer j specifically requests a subcompact and all subcompacts have been rented out, the agency may decide to give him a midsize for the price of a subcompact in order not to lose him as a customer. The set M_j includes subcompacts as well as midsizes (even though customer j requested a subcompact), but the agency's daily profit is a function of the car as well as of the customer, i.e., π_{ij} dollars per day, since the agency gives him a larger car at a lower price. The weight is $w_{ij} = \pi_{ij} p_j$. □

Most reservation problems can be formulated as integer programs. Time is divided in periods or slots of unit length. If the number of slots is fixed, say H, then the problem is referred to as an H-slot problem. Assume, for the time being, that the activity durations are equal to 1 and let J_t denote the set of activities that need a resource in slot t, i.e., during period $[t-1, t]$. If x_{ij} denotes a binary variable that assumes the value 1 if activity j is assigned to resource i and 0 otherwise, then the following constraints have to be satisfied:

$$\sum_{i=1}^{m} x_{ij} \le 1 \qquad j = 1, \ldots, n$$

$$\sum_{j \in J_t} x_{ij} \le 1 \qquad i = 1, \ldots n,\ t = 1, \ldots H .$$

The first set of constraints ensures that every activity is assigned to at most one resource and the second set ensures that a resource is not assigned to more than one activity in any given slot.

The easiest version of the reservation problem is a feasibility problem: does there exist an assignment of activities to resources with every activity being assigned to a resource? A slightly harder version of this feasibility problem would be the following: does there exist an assignment of activities to resources with activity j being assigned to a resource belonging to a given subset M_j? It turns out that this problem is still relatively easy.

In the optimization version of the reservation problem the objective is to maximize the total profit

$$\sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} x_{ij} ,$$

where the weight w_{ij} is equivalent to a profit associated with assigning activity j to resource i. Some special cases of this optimization problem can actually be solved in polynomial time. For example, consider again the case with all n activities having a duration equal to 1, i.e., $p_j = 1$ for all j, and assume arbitrary resource subsets M_j and arbitrary weights w_{ij}. Each time slot can be considered as a separate subproblem that can be solved as an independent assignment problem.

Another version of the reservation model that allows for an efficient solution assumes arbitrary durations, identical weights (i.e., $w_{ij} = 1$ for all i and j), and each set M_j consisting of all m resources (i.e., the m resources are identical). The durations, the starting times (release dates) and the completion times (due dates) are arbitrary integers and the objective is to maximize the number of activities assigned. This problem cannot be decomposed into a number of independent subproblems (one for each time slot), since the durations of the different activities may overlap. However, it can be shown that the following relatively simple algorithm maximizes the total number of activities. In this algorithm, which is due to Bouzina and Emmons (1996), the activities are ordered in increasing order of their release dates, i.e.,

$$r_1 \le r_2 \le \ldots \le r_n .$$

Set J denotes the set of activities already considered.

Algorithm 4.2.2: Maximizing Number of Activities Assigned

Step 1

Set $J = \varnothing$ and $j = 1$.

Step 2

If a resource is available at time r_j, then assign activity *j* to that resource; include activity *j* in *J*, and go to Step 4.

Otherwise go to Step 3.

Step 3

Let j^* be such that

$$C_{j^*} = \max_{k \in J}(C_k) = \max_{k \in J}(r_k + p_k) .$$

If $C_j = r_j + p_j > C_{j^*}$, do not include activity j in J and go to Step 4.

Otherwise, delete activity j^* from *J*, assign activity j to the resource freed and include activity *j* in *J*.

Step 4

If $j = n$, STOP, otherwise set $j = j + 1$ and return to Step 2.

The structure behind the algorithm above is actually quite typical of the scheduling optimization problems of this type. A schedule is being built up going forward in time. Whenever a selection has to be made between various alternatives, then the alternative is selected which results in a situation that is the least restrictive and the most favorable (e.g., provides the most freedom) for the remaining activities to be scheduled.

Another version of this reservation model with zero slack, arbitrary durations, equal weights, and identical resources is also of interest. Assume there are an unlimited number of identical resources in parallel and all activities have to be assigned. However, the assignment must be done in such a way that a minimum number of resources is used. This problem is, in a sense, a dual of the problem discussed before. It turns out that minimizing the number of resources when all activities have to be done is also an easy problem.

It can be solved as follows. Again, the activities are ordered in increasing order of their release dates, i.e., $r_1 \le r_2 \le \ldots \le r_n$. First, activity 1 is assigned to resource 1. The algorithm then proceeds with assigning the activities, one by one, to the resources. Suppose that the first $j-1$ activities have been assigned to resources $1, 2, \ldots, i$. Some of these activities may have been assigned to the same resource. So $i \le j-1$. The algorithm then takes the next activity from the list, activity *j*, and tries to assign it to a resource that already has been utilized before. If this is not possible, i.e., resources $1, 2, \ldots, i$ are all busy at time r_j, then the algorithm assigns activity *j* to resource $i+1$. The number of resources utilized after activity *n* has been assigned is the minimum number of resources required.

This last problem, with the activities having arbitrary durations, turns out to be a special case of the same well-known node coloring problem described in the timetabling section of this survey. Consider the *n* nodes and let node j correspond now to activity j. If there is an (undirected) arc (*j*, *k*) connecting nodes j and *k*, then the processing of activities *j* and *k* overlap in time and nodes *j* and *k* cannot be given the same color. If the graph can be colored with *m* (or less) colors, then a feasible schedule exists with m resources. This node coloring problem, which is a feasibility problem that is NP-hard, is actually more general than the reservation problem considered in this section in which the number of resources used is minimized.

That the reservation problem considered in this section (with the activities having arbitrary durations and the number of resources to be minimized) is not equivalent to the timetabling problem (with heterogeneous operators and all processing times equal to 1) but rather a special case can be shown as follows: two activities that need the same operator in the timetabling problem are equivalent to two activities that have an overlapping time slot in the reservation problem. If two activities in the reservation problem have an overlapping time slot, then the two nodes are connected. Each color in the coloring process represents a resource and minimizing the number of colors is equivalent to minimizing the number of resources in the reservation problem. That the reservation problem is a special case follows from the fact that the time slots required by an activity in a reservation problem are *adjacent*. However, it may not be possible to order the operators in the timetabling problem in such a way that the operators required for each activity are adjacent to one another. It is this adjacency property that makes the reservation problem easy, while the lack of adjacency makes the timetabling problem with operator constraints hard.

In this subsection we have only discussed reservation models without any slack, i.e., $p_j = d_j - r_j$. Of course, in practice, reservation systems are designed in such a way that a limited amount of slack is allowed, i.e., $p_j < d_j - r_j$. Typically, even though a limited amount of slack enables a reservation system to generate a better and more profitable solution, the optimization problems involved are often considerably harder than the optimization problems for reservation systems that do not allow for any slack.

A fair amount of research has been done on interval scheduling, often just in the form of single machine and parallel machine scheduling with release dates and due dates or deadlines; see, for example, Garey, Johnson, Simons and Tarjan (1981), Martel (1982a, 1982b), and Posner (1985). Martin, Jones and Keskinocak (2003) consider a very interesting reservation system for On-Demand Aircraft schedules for fractional aircraft operators.

4.3 Appointment Scheduling

The scheduling of appointments is a common practice in many service industries, mainly to utilize resources efficiently and to avoid queueing. Many papers have appeared in the literature on appointment scheduling, mostly motivated by health care applications. Cayirli and Veral (2003), Gupta and Denton (2008) provide overviews of the literature, the research challenges and opportunities. Hall (2012) provides a comprehensive review of models and methods used for scheduling the delivery of patient care for all parts of the health care system. The analysis may be based on anyone of a variety of approaches, including stochastic programming, queueing theory, and stylized scheduling models.

Appointment scheduling systems are widely used as a tool for managing patient arrivals at health care facilities in order to match supply with demand. In practice it is actually fairly common for patients not to show up for their scheduled services. Missed appointments result in under-utilization of valuable resources and limit the access for other patients who could have filled the empty slots. Meanwhile, patients nationwide experience difficulties in accessing medical appointments in a timely manner due to long backlogs. Poor appointment utilization and excessive delays for outpatient care are widely recognized as significant barriers to effective health care delivery.

Appointment overbooking is one operational strategy employed by health care providers to address the issue of no-shows and at the same time increase patients' access to care. However, overbooking may potentially result in an overcrowded facility, with increased patients' waiting times and system's overtime. Recent studies have demonstrated that a sensible practice of appointment overbooking can significantly improve the operational performance of a medical facility with patients enjoying shorter waiting times and better access to services, see for example LaGanga and Lawrence (2012), Robinson and Chen (2010), Zacharias and Pinedo (2014a), Zacharias and Pinedo (2014b).

In the case of homogeneous patients it is of interest to determine the number of patients to schedule every day and how to allocate these patients to the different slots. The sequencing of the patients is also of interest when patients have different characteristics (no-show rates, processing times, waiting cost coefficients). In most cases, finding an optimal schedule is analytically intractable, and thus, the majority the literature uses enumeration, search algorithms, simulation-based techniques and/or heuristics.

Outpatient clinics typically start empty at the beginning of a working day, operate for a finite amount of time (in the order of say 8-12 hours), and shut down until the next period. Therefore, it is important to perform transient analysis for the random evolution of such systems. As pointed out in Bandi and Bertsimas (2012), transient queues are difficult to analyze via classical queueing techniques. Typically the analysis of rich queueing systems over finite time horizons is addressed either by computer simulation or diffusion approximations.

The majority of the literature focuses on single-server models. Kaandorp and Koole (2007), Hassin and Mendel (2008), Klassen and Yoogalingam (2009), Robinson and Chen (2010), Millhiser and Veral (2014) consider the appointment scheduling problem with homogeneous patients who arrive on time for their scheduled appointments, if they do show up. Begen and Queyranne (2011), Cayirli et al. (2012), LaGanga and Lawrence (2012), Zacharias and Pinedo (2014a) account further for patient heterogeneity. Even though the literature for the single server system is quite extensive, the multi-server case has received limited attention. As pointed out by Gupta and Wang (2012) as well, appointment scheduling models become intractable if multiple features are considered simultaneously. A stylized scheduling model with $s \geq 1$ servers appears in Zacharias and Pinedo (2014b). Multi-server systems can be used to model for example a diagnostic facility where it is crucial to utilize resources (e.g. CT scan, X-ray generator, MRI) efficiently. Doctors are modeled as "parallel servers" in settings where *continuity of care* is not a big concern. Nurses are modeled as "parallel servers" when they are the bottleneck resource, and/or the presence of a doctor is not required (e.g. vaccination and immunization).

Consider the following simple, tractable, stylized scheduling model that provides useful insights into appointment scheduling. Variations of this model have been analyzed under different scopes by various papers in the literature including Robinson and Chen (2010), LaGanga and Lawrence (2012), Zacharias and Pinedo (2014a), Zacharias and Pinedo (2014b).

Consider $s \geq 1$ identical service providers working in parallel. Each one has in her regular schedule n time slots available to serve patients in a working day. Beyond these n regular slot, each one can serve patients in overtime slots as well. Arrivals are driven by scheduled appointments. Let m denote the number of patients to be scheduled throughout the working day, subject to optimization, and let $y = m - ns$ denote the level of overbooking. The scheduler would like to assign each one of the patients to arrive at the beginning of one of the time slots. Patients show up with probability $p = 1 - q$ at the beginning of the time slot they were assigned and require one time slot of service.

There are three costs associated with an appointment schedule: patients' waiting cost, servers' idle time and overtime costs. The objective is to minimize the weighted sum of these three costs. If there are less than s patients present at the beginning of any one of the regular n time slots, then one or more providers remain idle and for each provider being idle an idle time cost c_I is incurred. The scheduler may overbook certain time slots and assign more than s patients in order to compensate for the no-show behavior. If more than s patients are present at the beginning of a time slot due to overbooking, then all but s of these patients have to wait. A waiting cost w is incurred for each time slot that a patient has to wait before starting her service. An overtime cost c_O is incurred for each overtime slot that a provider has to remain present at the medical facility to serve patients at the end of the regular working day.

A schedule is denoted by a vector $\mathrm{x} = (x_1, \ldots, x_n)$, where x_t is the number of patients assigned to slot t, with $\sum_{t=1}^{n} x_t = m$. It can be shown that if x^* is an optimal schedule, then $x_t^* \geq s$ for all $t = 1, 2, \ldots, n$.

The backlog of patients at the beginning of slot t, denoted by L_t, satisfies the Lindley recursion

$$L_t = \max\left\{L_{t-1} + A_{t-1} - s, 0\right\}, \text{ for } t \geq 2,$$

and

$$L_1 = 0,$$

where $A_t \sim \text{Binomial}(x_t, p)$ denotes the number of new arrivals at slot t.

Let $f(k; n, p)$ be the probability that a $\text{Binomial}(n, p)$ random variable takes a value equal to k, i.e.,

$$f(k; n, p) = \binom{n}{k} p^k (1-p)^{n-k}$$

and let $\pi_t^j(\mathrm{x}) = \Pr(L_t = j)$ denote the probability of a backlog of j patients at the beginning of slot t under schedule x. Let also $l_t = \sum_{\tau=1}^{t}(x_\tau - s)$ denote the maximum possible backlog at the beginning of slot t. Assuming that the system is empty at the beginning of the working day, then $\pi_1^0(\mathrm{x})=1$ and $\pi_t^j(\mathrm{x})$ can be expressed recursively for $t = 2, 3, \ldots, n+1$ as

$$\pi_t^i(\mathrm{x}) = \begin{cases} \sum_{j=0}^{\min(s, l_{t-1})} \pi_{t-1}^j(\mathrm{x}) \sum_{k=0}^{s-j} f(k; x_{t-1}, p) & \text{for } i = 0 \\ \sum_{j=\max(0, s+i-x_t)}^{\min(s+i, l_{t-1})} \left(\begin{array}{c} \pi_{t-1}^j(\mathrm{x}) \\ f\left(\begin{array}{c} s+i-j; \\ x_{t-1}, p \end{array} \right) \end{array} \right) & \text{for } 1 \leq i \leq l_t \\ 0 & \text{otherwise.} \end{cases}$$

The expected system's overtime, idle time, and patients' aggregate waiting time can be expressed respectively as

$$O(\mathrm{x}) = E(L_{n+1}) = \sum_{j=0}^{l_{n+1}} j\pi_{n+1}^j(\mathrm{x}),$$

$$I(\mathrm{x}) = O(\mathrm{x}) + ns - pm,$$

$$W(\mathrm{x}) = \sum_{t=1}^{n} \sum_{i=1}^{x_t} \sum_{j=\max 0,(s-i+1)}^{l_t} \pi_t^j(\mathrm{x}) \sum_{k=\max(0,s-j)}^{i-1} \left(\begin{array}{c} rf(k; i-1, p) \\ \left\lceil \frac{j+k-s+1}{s} \right\rceil \end{array} \right).$$

Note that the second equation follows from the fact that (Number of servers' idle slots) + (Number of patients who show up) = *ns* + (Number of overtime slots). We consider the following nonlinear integer program

$$\min_{(\mathrm{x}, m)} \quad c_I I(\mathrm{x}) + \mathrm{c}_O O(\mathrm{x}) + wW(\mathrm{x})$$

$$\text{s.t.} \quad m \geq ns$$

$$x_t \geq s \quad t = 1, 2, \ldots, n$$

$$\sum_{t=1}^{n} x_t = m,$$

with $m, x_1, x_2, \ldots x_n$ being integer.

Optimal schedules with overbooking are *front loaded*: more patients are scheduled towards the beginning of the working day (in order to get an empty system running), and the schedules tend to become somewhat less dense towards the end of the working day (clearly in order to avoid high overtime costs). An example of such a schedule is displayed in Figure 1 for a system with 2 servers, a working day of 24 times slots (for example an 8-hour working day with 20-minute slots), and a no-show rate of 20%. For a more comprehensive numerical analysis the reader is referred to Zacharias and Pinedo (2014b). It is evident, and intuitive, that the optimal overbooking level is increasing in the no-show rate *q*. As *w* increases, the optimal schedules become less front-loaded, without necessarily observing a decrease in the overbooking level. Overbooking increases significantly with the number of parallel servers, and that increase is more prevalent for higher no-show rates.

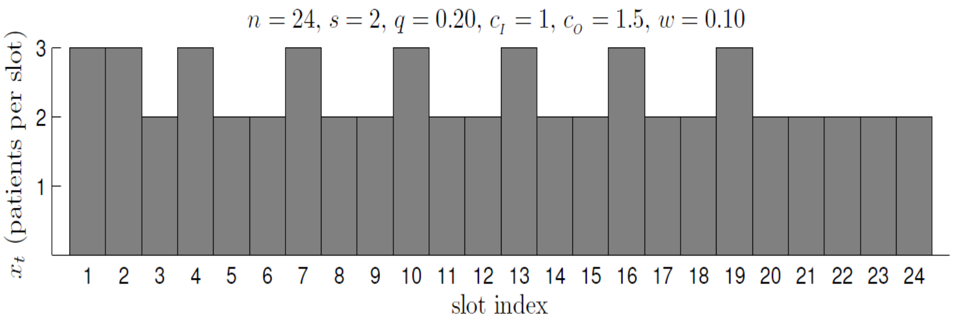

Figure 1. One day schedule profile

5. Transportation Scheduling

Transportation is a quintessential service that can take many different forms, dependent upon the mode of transportation. The various modes of transportation include buses, trains, airplanes and ships. The different modes of transportation have different planning horizons, are subject to different sets of constraints and have different objective functions. Each mode has its own set of scheduling techniques. For a handbook on transportation in general, see Barnhart and Laporte (2006).

5.1 Urban Transit Scheduling

Transit systems play a very important role in urban transportation systems. There is an intense competition between public transit systems and the use of private vehicles. An efficient public transit system may encourage individuals not to use their cars but rather use a public bus or subway system. This would mitigate traffic congestion and reduce pollution.

Efficient planning and scheduling of urban transit systems in general would improve the performance of transit systems. There are typically four phases in the planning and scheduling of urban transit systems, namely

(i) planning of the construction of the transit network,

(ii) timetabling design of the bus or subway system,

(iii) vehicle/train scheduling, and

(iv) crew scheduling.

Of these four phases, three are scheduling related. The transit network design is an exception. The objective of the transit network design problem is to minimize the costs of the various resource investments under fixed or variable traffic demand. The optimization problem is constrained by the selection of the routes and the bus/train capacities. After the design of the transit network has been fixed, the timetables for each one of the bus routes have to be determined.

Bus timetables clearly depend on traffic demand. Departure and arrival times for each trip on each line of the entire transit network have to be determined. Once the departure and arrival times have been determined, the headway and frequencies of the buses are also known. The most important objective in the bus timetabling problem is the minimization of the total waiting time of the passengers. Timetabling problems become quite interesting when transit networks are considered. In transit networks, transfers are very important and have to be taken into account. Therefore, timetabling requires a synchronization of buses in order to minimize passengers' waiting times at transfer nodes and avoid "bunching" of buses.

In what follows we use the notation from Ceder et al. (2001). A bus network is denoted by $G=\{A, N\}$, where A represents the set of bus routes and N represents the set of transfer nodes. Let T denote the time horizon, M the number of bus routes, N the number of transfer nodes, and $H_{\min k}$ and $H_{\max k}$ are the minimum and maximum headway allowed between two consecutive departures on route k. Let F_k denote the total number of departures on route k, T_{kj} the travel time between the starting point and node j on route k. The decision variables are X_{ik} and I_{ikjln}; the X_{ik} represent the departure time of the ith bus on route k and $Z_{ikjqn}=1$ implies that the arrival time at node n of the ith bus on route k is the same as the arrival time of the jth bus on route l at node n. The objective is to maximize the number of synchronized trips among all departures. The objective function is therefore:

$$\sum_{k=1}^{M-1}\sum_{i=1}^{F_k}\sum_{q=k+1}^{M}\sum_{j=1}^{F_q}\sum_{n\in A_{kl}} I_{ikjln}$$

where A_{kl} is the set of shared nodes between routes k and l.

Bus scheduling requires that bus headways have to lie in between certain minimum and maximum values. A headway may be neither too big nor too small. A headway that is too big may increase the waiting times of passengers while a headway that is too small may result in high operational costs, running times with small numbers of passengers, and higher probabilities of bunching of buses.

A solution to a bus scheduling problem specifies the departure and arrival times on each route. Based on the assumption, adopted in most studies, that the travel times on specific bus lines are deterministic, a trip can be defined by the departure and arrival times.

The problem now becomes how to assign vehicles to the respective trips, see Freling et al. (2001). If we have only one depot for all the vehicles, we have a so-called single-depot vehicle scheduling problem. The input for such a problem is the location of the depot and the set of trips with their departure and arrival times. Furthermore, a travel time matrix is given which provides all the travel times between all the locations. A feasible solution for the vehicle scheduling problem requires that each trip has to be assigned to a vehicle and each vehicle operates a set of consecutive trips. The vehicle starts out from the depot; after completing all trips, the vehicle returns to the depot. Typical criteria in the search for an optimal vehicle assignment involves fixed and operational costs.

In the vehicle scheduling context, a new network is designed in which the nodes represent the trips and the arcs connect trips that are "compatible". The term compatible implies that the ending time of one trip is earlier than the starting time of the trip immediately following. The network is denoted by $G=(A',N')$. Vehicle movements between two consecutive trips are referred to as idle times without servicing passengers. Two dummy nodes are added to the network, both representing the depot; the two dummy nodes are denoted by s_1

and s_2. A feasible vehicle schedule consists of several consecutive trips that are compatible starting with trip s_1 and ending with trip s_2. The c_{ij} is the cost associated with a vehicle movement from trip i to trip j and is a known parameter. The decision variable y_{ij} represents the relationship between successive trips.

The vehicle scheduling model can now be formulated as follows:

$$\min \sum_{(i,j)\in A'} c_{ij} y_{ij}$$

$$\sum_{j:(i,j)} y_{ij} = 1, \quad i \in N'$$

$$\sum_{i:(i,j)\in A'} y_{ij} = 1, \quad j \in N'$$

$$y_{ij} \in \{0,1\}, \ (i,j) \in A'$$

The solution of the vehicle scheduling model offers a set of disjoint paths from s_1 to s_2.

A natural extension of the single-depot vehicle scheduling problem is the Multidepot Vehicle Sheduling Problem (MDVSP). In the MDVSP, each vehicle belongs to a given depot and each trip is assigned to only one vehicle. Models for the MDVSP can be classified as single-commodity flow models, multicommodity flow models, and set partitioning models. The methods used for solving such models include Branch-and-Cut, LP-Relaxation methods, and column generation methods, see Fischetti et al., (2001), Mesquita et al., (1999), Ribeiro et al., (1994).

Crew scheduling takes place after the vehicle scheduling problem has been solved. Several concepts have to be introduced before the crew scheduling model can be formulated. A vehicle block is defined as a vehicle movement from one depot to another. There are one or more trips between the departure and end depot. A deadhead represents an inevitable movement of a vehicle in between two trips or in between a trip and a depot without servicing passengers. Relief points are points that contain time as well as space information regarding a driver being able to have a break or being able to leave. A task is a sequence of trips and deadheads in between two relief points which represents the smallest unit of work that can be assigned to a crew member. A piece is a sequence of consecutive tasks without breaks in between which is also called a duty. The objective in the crew scheduling problem is to minimize the total cost of all duties while each task is part of at least one duty. Recently, the simultaneous optimization of vehicle and crew scheduling have been studied as well, see Huisman et al., (2005). Simultaneous optimization involving both vehicle and crew scheduling usually results in significant cost savings, see Haase et al. (2001).

In the field of urban transportation scheduling the timetabling problems for Mass Rapid Systems (MRT) or Metro systems are similar to train timetabling problems. Most studies in the current literature have focused on a single, one way track that connects two major stations with several smaller stations in between. An express train may not stop at a smaller station in between. A controller can slow down a train or make the dwelling time at an intermediate station longer. Let $S = \{1, 2, \ldots, s\}$ denote the set of stations and let $T = \{1, 2, \ldots, t\}$ denote the set of trains. The decision variables in the train timetabling problem are the departure times at stations $\{1, 2, \ldots, s-1\}$ and the arrival times at stations $\{2, 3, \ldots, s\}$ for each train t.

There is actually an ideal timetable for each train; the ideal timetable is given and depends on passenger behavior and preferences. The most popular objective function for the train timetabling problem is to minimize the cost that is associated with the deviation of the actual timetable from the ideal timetable. Constraints in the train timetabling problem include track capacity constraints, time window constraints and other physical constraints. The most popular objective function for the train timetabling problem is to minimize the cost that is associated with the deviation of the actual timetable from the ideal timetable. Constraints in the train timetabling problem include track capacity constraints, time window constraints and other physical constraints.

Bus as well as train timetabling are supposed to generate the arrival as well as the departure times of each bus (train) at each stop (station). Most of the differences between bus timetabling and train timetabling are due to the existence of tracks; some of these differences are the following:

(i) Because of safety regulations, a minimum headway has to be maintained between two consecutive trains. However, for bus systems the headways are not a hard constraint.

(ii) Takeover phenomena in the context of train timetabling occur only at the train stations. However, bus timetabling does not have any constraints with regard to takeovers.

(iii) Controllers can manipulate both the speed of trains and the dwelling times at the train stations. The control granularity of trains is higher than of buses. In the context of bus operation, it is impossible to control the running time and dwell time for a bus since traffic in urban transportation networks is highly uncertain.

Mainly because of these differences, the modeling of the train timetabling problem is more complicated than that of the bus timetabling problem, see Carey and Lockwood (1995), and Caprara et al. (2002). Bus timetabling typically faces the challenges of a highly uncertain urban transportation environment. All these uncertainties (e.g. uncertain on-route travel time, uncertain dwell time on bus stops and etc.) are hard to model.

An extensive amount of research has been done on the scheduling of public transport (buses and trains). This has resulted in a number of proceedings of conferences, see Wren and Daduna (1988), Desrochers and Rousseau (1992), Daduna et al. (1995), Wilson (1999).

5.2 Maritime Scheduling

Scheduling plays a very important role in shipping. An enormous amount of work has been done on scheduling in this mode of transportation. Historically, Mathematical Programming and in particular Integer Programming has played a very important role in maritime as well as in aviation scheduling. There are several special types of Integer Programming formulations that have numerous practical applications in many of the various forms of transportation scheduling. This subsection first describes three very widely used Integer Programming formulations, namely Set Partitioning, Set Covering, and Set Packing. The integer programming formulation of the Set Partitioning problem has the following structure.

$$\text{Minimize} \quad c_1 x_1 + c_2 x_2 + \ldots + c_n x_n$$

subject to

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = 1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = 1, \\ \qquad\qquad\qquad\qquad \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = 1, \\ x_j \in \{0,1\} \qquad \text{for } j = 1,\ldots,n, \end{cases}$$

with all a_{ij} values being either 0 or 1.

When the equal signs ($=$) in the constraints are replaced by greater or equal signs ($\geq$), the problem is referred to as the Set Covering problem, and when the equal signs are replaced by less than or equal signs ($\leq$), then the problem is referred to as the Set Packing problem. In practice, the objective of the Set Packing problem is typically a profit maximization objective.

The mathematical model underlying the Set Partitioning problem can be described as follows: Assume m different elements and n different subsets of these m elements. Each subset contains one or more elements. If $a_{ij} = 1$, then element i is part of subset j, and if $a_{ij} = 0$, then element i is not part of subset j. The objective is to find a collection of subsets such that each element is part of exactly one subset. The objective is to find that collection of subsets that have a minimum cost. In the Set Covering problem, each element has to be part of at least one subset. In the Set Packing problem each subset yields a certain profit π_j and the total profit has to be maximized in such a way that each element is part of at most one subset.

A well-known maritime scheduling example is the so-called Tanker Scheduling Problem, which is an example of the Set Packing problem mentioned before. The reason why the tanker scheduling problem is a Set Packing problem and not a Set Partitioning problem is based on the fact that not every cargo has to be transported by a company owned tanker. If it is advantageous to assign a cargo to an outside charter, then that is allowed. This implies that the first set of constraints in the MIP are inequality ($\leq$) constraints rather than equality (=) constraints.

Oil companies that own and operate tanker fleets typically make a distinction between two types of ships. One type of ship is company-owned and the other type of ship is chartered. The operating cost of a company-owned ship is different from the cost of a charter that is typically determined on the spot market. Each ship has a specific capacity, a given draught, a range of possible speeds and fuel consumptions, and a given location and time at which the ship is ready to start a new trip.

Each port also has its own characteristics. Port restrictions take the form of limits on the deadweight, draught, length, beam and other physical characteristics of the ships. There may be some additional government rules in effect as well.

A cargo that has to be transported is characterized by its type (e.g., type of crude), quantity, load port, delivery port, time window constraints on the load and delivery times, and the load and unload times. A schedule for a ship defines a complete itinerary, listing the sequence of ports to be visited within the time horizon, the time of entry at each port and the cargoes loaded or delivered at each port.

The objective is to minimize the total cost of transporting all cargoes. This total cost consists of a number of elements, namely the operating costs for the company-owned ships, the spot charter rates, the fuel costs, and the port charges. Port charges vary greatly between ports and within a given port charges typically vary proportionally with the deadweight of the ship.

In order to present a formal description of the problem the following notation is used. Let n denote the number of cargoes to be transported, T the number of company-owned tankers, and p the number of ports. Let S_i denote the set of all possible schedules for ship i. Schedule l for ship i, $l \in S_i$, is represented by the column vector

$$\begin{matrix} a_{i1}^l \\ a_{i2}^l \\ \vdots \\ a_{in}^l \end{matrix}$$

The constant a_{ij}^l is 1 if under schedule l ship i transports cargo j and 0 otherwise. Let c_i^l denote the incremental cost of operating a company-owned ship i under schedule l versus keeping ship i idle over the entire planning horizon. The operating cost can be computed once schedule l has been specified, since it may depend in various ways on the characteristics of the ship and of the schedule, including the distance travelled, the time the ship is used, and the ports visited. The cost c_j^* denotes the amount that has to be paid on the spot market to transport cargo j on a ship that is not company owned.

Let

$$\pi_i^l = \sum_{j=1}^{n} a_{ij}^l c_j^* - c_i^l$$

denote the "profit" (i.e., the amount of money that does not have to be paid on the spot market) by operating ship i according to schedule l. The decision variable x_i^l is 1 if ship i follows schedule l and zero otherwise.

The Tanker Scheduling Problem can now be formulated as follows:

$$\text{Maximize} \quad \sum_{i=1}^{T} \sum_{l \in S_i} \pi_i^l x_i^l$$

subject to

$$\sum_{i=1}^{T} \sum_{l \in S_i} a_{ij}^l x_i^l \le 1 \qquad j = 1, \ldots, n$$

$$\sum_{l \in S_i} x_i^l \le 1 \qquad i = 1, \ldots, T$$

$$x_i^l \in \{0,1\} \qquad l \in S_i, \ i = 1, \ldots, T$$

The objective function specifies that the total profit has to be maximized. The first set of constraints imply that each cargo can be assigned to at most one tanker. The second set of constraints specifies that each tanker can be assigned at most one schedule. The remaining constraints imply that all decision variables have to be binary $0-1$. This optimization problem is typically referred to as a set-packing problem.

The algorithm used to solve this problem is a branch-and-bound procedure. However, before the branch-and-bound procedure is applied, a collection of candidate schedules have to be generated for each ship in the fleet. As stated before, such a schedule specifies an itinerary for a ship, listing the ports visited and the cargoes loaded or delivered at each port. The generation of an initial collection of candidate schedules has to be done by a separate ad-hoc heuristic that is especially designed for this purpose. The collection of candidate schedules should include enough schedules so that potentially optimal schedules are not ignored, but not so many that the set-packing problem becomes intractable. Physical constraints such as ship capacity and speed, port depth and time windows limit the number of feasible candidate schedules considerably. Schedules that have a negative profit coefficient in the objective function of the set-packing formulation can be omitted as well.

The branch-and-bound method for solving the problem is typically based on customized branching and bounding procedures. Since the problem is a maximization problem a good schedule generated by a clever heuristic (or a manual method) provides a lower bound for the value of the optimal solution. When considering a particular node in the branching tree, it is necessary to develop an upper bound for the collection of schedules that correspond to all the descendants of this particular node; if this upper bound is less than the lower bound on the optimum provided by the best schedule currently available, then this node can be fathomed.

There are a variety of suitable branching mechanisms for the branch-and-bound tree. The simplest mechanism is just the most basic $0-1$ branching. Select at a node a variable x_i^l which has not been fixed yet at a higher level node and generate branches to two nodes at the next level down: one branch for $x_i^l = 0$ and one for $x_i^l = 1$. The selection of the variable x_i^l may depend on the solution of the LP relaxation at that node; the most suitable x_i^l may be the one with a value closest to 0.5 in the solution of the LP relaxation. If at a node a variable x_i^l is set equal to 1 for ship *i*, then certain schedules for other ships can be ruled out for all the descendants of this node; that is, the schedules for other ships that have a cargo in common with schedule *l* for ship *i* do not have to be considered any more.

Another way of branching can be done as follows: Select at a given node a ship *i* that has not been selected yet at a higher level node and generate for each schedule *l* in S_i a branch to a node at the next level down. In the branch corresponding to schedule *l* the variable $x_i^l = 1$. Using this branching mechanism, one still has to decide at each node which ship *i* to select. One could select the *i* based on several criteria. For example, a ship that transports many cargoes or a ship that may be responsible for a large profit. Another way is to select an *i* that has a highly fractional solution in the LP relaxation of the problem (e.g., there may be a ship *i* with a solution $x_i^l = 1/K$ for *K* different schedules with *K* being a fairly large number).

An upper bound at a node can be obtained by solving the linear relaxation of the set-packing problem corresponding to that node, i.e., the integrality constraints on x_i^l are replaced by the non-negativity constraints $x_i^l \geq 0$. This problem may be referred to as the *continuous* set-packing problem. The value of the optimal solution is an upper bound for the values of all possible solutions of the set-packing problem at that node. It is nowadays possible to find with little computational effort optimal solutions (or at least good upper bounds) for very large continuous set-packing problems, making such a bounding mechanism quite effective.

For details regarding tanker scheduling or scheduling in shipping in general, see Brown, Graves and Ronen (1987), Fisher and Rosenwein (1989), Perakis and Bremer (1992), Christiansen (1999), Christiansen et al. (2004), Christiansen et al. (2006).

5.3 Aircraft Scheduling

Aircraft scheduling provides an interesting contrast to the tanker scheduling just considered.

The classical aircraft routing and scheduling problem is an example of the standard Set Partitioning problem. It is clear that in the aviation industry the constraint that each flight leg should be covered *exactly* once by a round trip is important. On the other hand, in the trucking industry, it may be possible to have one leg covered by several round trips; the constraints may then be relaxed and the problem assumes a Set Covering structure.

The Daily Aircraft Routing and Scheduling Problem can now be described as follows: Given a heterogeneous aircraft fleet, a collection of flight legs that have to be flown in a one-day period with departure time windows, durations, and cost/revenues corresponding to the aircraft type for each leg, a fleet schedule has to be generated that maximizes the airline's profits (possibly subject to certain additional constraints).

Some of the additional constraints that often have to be taken into account in an Aircraft Routing and Scheduling Problem are the number of available planes of each type, the restrictions on certain aircraft types at certain times and at certain airports, the required connections between flight legs (the so-called "thrus") imposed by the airline and the limits on the daily service at certain airports. Also, the collection of flight legs may have to be balanced, i.e., at each airport there must be, for each airplane type, as many arrivals as departures. One must further impose at each airport the availability of an equal number of aircraft of each type at the beginning and at the end of the day.

In the formulation of the problem the following notation is used: L denotes the set of flight legs, T denotes the number of different aircraft types, and m_i denotes the number of available aircraft of type i, $i = 1,\ldots,T$. So the total number of aircraft available is

$$\sum_{i=1}^{T} m_i \ .$$

Some fight legs may be flown by more than one type of aircraft. Let L_i denote the set of flight legs that can be flown by an aircraft of type i and let S_i denote the set of feasible schedules for an aircraft of type i. This set includes the empty schedule (0); an aircraft assigned to this schedule is simply not being used. Let π_{ij} denote the profit generated by covering flight leg j with an aircraft of type i. With each schedule $l \in S_i$ there is a total anticipated profit

$$\pi_i^l = \sum_{j \in L_i} \pi_{ij} a_{ij}^l \ ,$$

where a_{ij}^l is 1 if schedule l covers leg j and 0 otherwise. If an aircraft has been assigned to an empty schedule, then the profit is π_i^0 . The profit π_i^0 may be either negative or positive. It may be negative when there is a high fixed cost associated with keeping a plane for a day; it may be positive when there is a benefit having a plane idle (some airlines want to have idle planes that can serve as stand by). Let A denote the set of airports, and A_i be the subset of airports that have facilities to accommodate aircraft of type i. Let o_{ih}^l be equal to 1 if the origin of schedule l, $l \in S_i$, is airport h, and 0 otherwise; let d_{ih}^l be equal to 1 if the final destination of schedule l is airport h, and 0 otherwise.

The binary decision variable x_i^l takes the value 1 if schedule l is assigned to an aircraft of type i, and 0 otherwise; the integer decision variable x_i^0 denotes the number of unused aircraft of type i, i.e., the aircraft that have been assigned to an empty schedule.

The Daily Aircraft Routing and Scheduling Problem can now be formulated as follows:

Maximize $\sum_{i=1}^{T} \sum_{l \in S_i} \pi_i^l x_i^l$

subject to

$$\sum_{i=1}^{T} \sum_{l \in S_i} a_{ij}^l x_i^l = 1 \qquad j \in L$$

$$\sum_{l \in S_i} x_i^l = m_i \qquad i = 1, \ldots, T$$

$$\sum_{l \in S_i} (d_{ih}^l - o_{ih}^l) \mathrm{x}_i^l = 0 \qquad i = 1, \ldots T, h \in A_i$$

$$x_i^l \in \{0,1\} \qquad i = 1, \ldots, T, l \in S_i$$

The objective function specifies that the total anticipated profit has to be maximized. The first set of constraints imply that each flight leg has to be covered exactly once. (This set of constraints is somewhat similar to the first set of constraints in the formulation of the tanker scheduling problem.) The second set of constraints specifies the maximum number of aircraft of each type that can be used. The third set of constraints correspond to the flow conservation constraints at the beginning and at the end of the day at each airport for each aircraft type. The remaining constraints imply that all decision variables have to be binary $0-1$. This model is basically a Set Partitioning Problem with additional constraints. The algorithm to solve this problem is also based on Branch-and-Bound; the version of Branch-and-Bound is typically referred to as Branch-and-Price.

The aircraft scheduling problem described above typically results in cyclic schedules. This is in contrast to the schedules for tankers (oil, natural gas, bulk cargo in general) which are usually not cyclic; the scheduling process for tankers is usually based on a rolling horizon procedure. For more details concerning scheduling in the aviation industry, see Stojkovich (2002), Desaulniers (1997), Barnhart et al. (1998), Cordeau et al. (2001), and Barnhart et al. (2003).

5.4 Emergency Operations Scheduling

Whenever a geographical area is hit by a disaster, emergency operations have to be organized. Disasters can take many different forms. A disaster may be an act of nature (e.g., hurricane, earthquake, etc.) or may be man-made (e.g., act of war, etc.). Some acts of nature may give an advance warning of a couple of days (e.g., hurricane), other acts of nature may not (e.g., earthquake, tsunami).

The management of operations before, during and after the occurrence of a disaster is clearly of importance. In general, there are four stages in emergency operations management; the four stages being mitigation, preparedness, response and recovery. In this section, we particularly focus on the response stage, since it is this stage that requires routing and scheduling. A response to a disaster involves the allocation of resources that are needed for mitigating the economic and human losses. In what follows, we focus on the planning and scheduling of the allocation of such resources. The planning and scheduling of the resource allocations refer mainly to the effective transportation of the different resources to the disaster areas. There are basically two different types of resources to be allocated, namely renewable resources (e.g., personnel, equipment) and non-renewable resources (e.g., medical supplies, building material). The allocation of the resources has to be coordinated in an efficient manner. We first concentrate on the resource planning and scheduling problem and assume that routing and transportation alternatives are known in advance and do not need to be optimized. However, assuming that the transportation routes are known in advance is often not reasonable and practical. The transportation routes in many situations are not determined in advance. Travel time of transportation is another critical factor that influences the routing decision significantly.

Before any rescue activities can be organized, it is necessary to build temporary emergency distribution centers (DCs) in order to accelerate the rescue processes. The emergency distribution centers (DCs) can take the form of warehouses that store the supplies, temporary hospitals, medical supply centers, and so on. The planning stage may require a formulation of models that can help determine the number and locations of such DCs. The most important differences between location problems in emergency scenarios and traditional location problems are the following:

(i) a low frequency of disaster occurrences;

(ii) a high frequency of requests for aid from the distribution centers (after a disaster, the demands from affected areas may be huge);

(iii) different demand areas having different characteristics (due to the diversity of population density, economic situation and other features of affected areas, the impact of a disaster on each potential demand area may be quite different);

(iv) resources have to be coordinated in order to accomplish rescue tasks;

(v) the level of uncertainty in the environment may be elevated.

An example of this uncertainty is the availability of transportation networks. Basically, there are three standard models that are often referred to as the covering models, namely the P-median and the P-center models, see Jia, Ordonez and Dessouky (2007). For a framework of emergency planning and scheduling, see Caunhye, Nie and Pokharel (2012).

Once the DCs have been set up, the problem boils down to on how to distribute the supplies in order to meet the demands. There are two important research directions here: one approach, which is more macroscopic, models the distribution of the emergency supplies as a multi-commodity network flow problem. The goods consists of renewable and non-renewable items that should be coordinated to mitigate the negative effects of disaster. The second approach, which is more microscopic, models the distribution problem as a vehicle routing problem. Let us first consider the network flow modeling paradigm and then go into the vehicle routing problem in an emergency context.

After the DCs have been set up, emergency materials have to be distributed in the affected areas. Commodity network flow models may be used. Quantities of commodity flows are going to be determined by the network flow models. The objectives include the minimization of the transportation cost, the makespan of the schedule, the amount of food delivered, the total amount of demand that has not been met, etc. The constraints include the number of vehicles and their capacities, the capacity of the links in the transportation network, and so on.

Commodity network flow models are not enough to depict the complexity of emergency resource scheduling. Disaster response scheduling has one very unique feature that requires the simultaneous scheduling and coordination of renewable resources (e.g., specialists, medical personnel, etc.) as well as of non-renewable resources (e.g., syringes, antibiotics, surgical blades, vaccines and bandages, etc.). When operations scheduling depends on both renewable and nonrenewable resources and the availability of both type resources have to be satisfied, the resulting scheduling problems tend to be very difficult.

In reality, renewable and nonrenewable resources have to be coordinated and synchronized with one another so that rescue activities can be performed. Renewable resources are also called permanent resources. In the resource assignment and project scheduling literature, some work has been done in this area, see Ait-Kadia et al. (2011), Wong et al. (2013) and Van Peteghem and Vanhoucke (2014); for a literature review on project scheduling, see Weglarz et al. (2011). In a typical emergency resource assignment and scheduling problem that takes both renewable and nonrenewable resources into account, the starting time of a service in an affected area can occur only after both the required renewable and nonrenewable resources have arrived. There are also some constraints with regard to the demand and the supplies. Another class of constraints deals with the travel times of the transportation. The objective function can be the tardiness of the services, see Lee and Lei (2001).

In the planning and scheduling of emergency operations, the routing of the vehicles plays an important role in the delivery system. The modeling of the delivery system is similar to a Vehicle Routing Problem (VRP). The traditional vehicle routing problem is one of the basic problems in the transportation and logistics domain. It focuses on the optimal routing design of the delivery of the goods at the distribution center to customers who are scattered at different locations.

Traditional vehicle routing problems can be formulated following the notation and formulation by Campbell, Vandenbussche, and Hermann (2008). Let $N = \{1, 2, \ldots, N\}$ denote the set of customers and let 0 denote the depot. Let t_{ij} denote the travel time between nodes i and j. The binary variable x_{ij} indicates whether or not a vehicle travels from node i to node j. Let a_i denote the arrival time of customer i. A vehicle routing problem in the emergency context can now be formulated as follows:

$$\text{Minimize} \quad \sum_{i,j \in N_0} t_{ij} x_{ij}$$

subject to

$$\sum_{j \in I_0} x_{ij} = 1, \quad i \in I_0$$

$$\sum_{j \in I_0} x_{ij} - \sum_{j \in I_0} x_{ji} = 0, \quad i \in I_0$$

$$t_{ij} + a_i \leq a_j + T(1 - x_{ij}), \quad i, j \in N$$

$$a_i \geq t_{0i} x_{0i}, \quad i \in N$$

$$x_{ij} \in \{0,1\}, \quad i, j \in N_0$$

$$a_i > 0, \quad i \in N_0$$

The objective of the mixed integer programming model is to minimize the total travel time. The constraints maintain conservation of flows and impose bounds on the travel times.

In the disaster management context, it is necessary to consider other factors by adding variables and constraints into the mathematical programming model. There are many other constraints that can be considered in a vehicle routing problem in a disaster management context which are the following:

(i) In a typical vehicle routing problem, the vehicles depart from a given depot and return to the depot after all service provisions have been taken care of. In an emergency situation, due to the high level of uncertainty, the emergency vehicle may not return to the depot, but stay at a stop on its route waiting for further instructions. Another possibility for this routing style is that vehicles need on the one hand dispatch goods to affected areas and on the other hand move sick and injured people to medical centers. The medical centers can also be in affected areas. The supplier and demand areas are interchangeable. This feature makes the new vehicle routing problem a bit like a simultaneous pickup-delivery vehicle routing problem.

(ii) The vehicle capacity may not be able to meet the demand. In such a case, multiple vehicles are needed to meet the demand at one of the affected areas. This situation is like the Split Delivery Vehicle Routing Problem (SDVRP). In SDVRP, the requirement that each customer can only be visited once no longer applies.

(iii) Unpredictability of demand. This uncertainty is different from traditional commercial delivery systems where typically demand is relatively stable, see Campbell, Vandenbussche and Herman (2008).

(iv) Coordination among organizations. There are various suppliers. Different vehicles are capable of delivering different resources. An example of this situation is that medical teams and medical suppliers need to be coordinated. The vehicles need to be coordinated as well.

(v) Some items are delivered only once during the entire disaster relief period and some items may need daily-based delivery such as food and water, particular in hot weather. For a survey paper that focuses on the periodic vehicle routing problem, see Francis, Smilowitz and Tzur (2008).

On the objective function side, it is important to deliver the goods to the demand areas in a fast and fair manner. Market forces and mechanisms tend to suddenly disappear in disaster relief operations. The disappearance of markets changes the nature of economics and operations. Even though there are no unified rules for allocating resources in disaster operations, fairness is a very important aspect of the goods delivery; transportation costs are not a top priority in disaster relief. Therefore, many objective functions have been developed with an emphasis on fairness. Some examples of such objective functions are:

(i) Minimize the arrival time of last visited area.

(ii) Minimize the average arrival time.

(iii) On the route level, fairness is measured as the weighted arrival time normalized by the demand on node, see Huang, Smilowitz and Balcik (2012).

(iv) Number of unsatisfied demands.

For a discussion with regard to the appropriateness of the various types of objective functions in disaster relief operations, see Holguin-Veras, Jaller, Wassenhove, Pérez and Wachtendorf (2012) and Holguin-Veras, Pérez, Jaller, Wassenhove and Aros-Vera (2013).

In order to solve the various different types of vehicle routing problems, a cornucopia of exact and heuristic algorithms have been developed. Baldacci, Mingozzi and Roberti (2012) present an overview of the exact algorithms for VRP under capacity and time window constraints. Augerat et al. (1998) developed a Branch-and-Cutalgorithm for solving the capacitated VRP. Set Partition formulations have also been applied to both the capacitated VRP and the VRP with time windows. The SD-VRP has also been used for modeling this dispatching problem in disaster operations settings. A two-stage algorithm with valid inequalities is proposed to solve the split delivery vehicle routing problem, see Jin, Liu and Bowden (2007). For a shortest path search algorithm based on dynamic programming for the vehicle routing problem with split pickups, see Lee et al. (2006). Normally, exact algorithms for solving the VRP and its large number of variants are usually constrained by the problem size. A number of heuristics have therefore been developed for dealing with large size problems: for tabu search algorithms for the Split Delivery Vehicle Routing Problem, see Archetti, Speranza and Hertz (2006) and Archetti, Speranza and Savelsbergh (2008).

6. Professional Sports and Entertainment

Professional sports and entertainment are very important service industries with their own sets of scheduling problems. They have their own planning horizons, their own objective functions and their own constraints.

6.1 Tournament Scheduling in Professional Sports

Sport events are a very important segment of the entertainment and recreation industry. Many different branches of sports maintain regular local, regional or national tournaments. There is a great variety in types of tournaments, each type having its own rules and restrictions. These tournaments have to be scheduled and it turns out that most of these schedules are anything but trivial to come by. This industry has inspired a significant amount of theoretical as well as more applied and more computational research, see Butenko et al. (2004). The more theoretical research established a very strong link between tournament scheduling and graph theory, see De Werra (1988).

Many tournament schedules are constrained in time; that is, the number of rounds or slots in which games are played is equal to the number of games each team must play plus some extra rounds or slots that are typically required in leagues with an odd number of teams. For example, in a so-called single round robin tournament each team has to play every other team once, either at home or away. Such a tournament among n teams with n being even requires $n-1$ rounds. If the number of teams is odd, then the number of rounds is n (due to the fact that in every round one of the teams has to remain idle). In a double round robin tournament each team has to play every other team twice, once at home and once away. It turns out that such a tournament among n teams requires either $2n-2$ or $2n$ rounds (dependent upon whether n is even or odd).

In order to formulate the most basic version of a tournament scheduling problem certain assumptions have to be made. Assume for the time being that the number of teams, n, is even. (It happens to be the case that tournament scheduling with an even number of teams is slightly easier to analyze than with an odd number of teams.) Consider a single round robin tournament in which each team has to play every other team exactly once, i.e., each team plays $n-1$ games. Because of the fact that there are an even number of teams it is possible to create for such a tournament a schedule that consists exactly of $n-1$ rounds with each round having $n/2$ games.

More formally, let t denote a round (i.e., a date or a time slot) in the competition. The 0 − 1 variable x_{ijt} is 1 if team i plays at home against team j in round t; the variable x_{ijt} is 0 otherwise. Of all the x_{ijt} variables a total of $(n/2)(n-1)$ are 1; the remaining are 0. The following constraints have to be satisfied:

$$\sum_{i=1}^{n}(x_{ijt}+x_{jit})=1 \quad \text{for} \quad j=1,\ldots,n; t=1,\ldots,n-1,$$

$$\sum_{t=1}^{n-1}(x_{ijt}+x_{jit})=1 \quad \text{for} \quad i \neq j.$$

In practice, there are usually many additional constraints concerning the pairing of teams and the sequencing of games. When there are a large number of constraints, one may just want to find a *feasible* schedule; finding a feasible schedule may already be hard.

However, it may at times also occur that one would like to optimize an objective. In order to formulate one of the more common objective functions in tournament scheduling some terminology is needed. If one considers the sequence of games played by a given team, each game can be characterized as either a *Home (H)* game or as an *Away (A)* game. The pattern of games played by a given team can thus be characterized by a string of H's and A's, e.g., *HAHAA*. There is typically a desire to have for any given team the home games and the away games to alternate. That is, if a team plays one game at home, it is preferable to have the next game away and vice versa. If a team plays in rounds t and $t+1$ either two consecutive games at home or two consecutive games away, then the team is said to have a break in round $t+1$. A common objective in tournament scheduling is to minimize the total number of breaks, see Trick (2001). It has been shown in the literature that in any timetable for a single round robin tournament with n teams (n being even), the minimum number of breaks is $n-2$. The algorithm that generates a schedule with this minimum number of breaks is constructive and very efficient; see Miyashiro and Matsui (2003).

Example 6.1.1: Breaks in a Single Round Robin Tournament

Consider 6 teams and 5 rounds.

	round 1	round 2	round 3	round 4	round 5
term 1	-6	3	-5	2	-4
term 2	-5	6	*4	-1	3
term 3	4	-1	*-6	5	-2
term 4	-3	5	-2	6	*1
term 5	2	-4	1	-3	*-6
term 6	1	-2	3	-4	5

When team i plays in round t against team j and the game is entered in the table as j, then team i plays at the site of team j. If it is entered as $-j$, then team i plays at home. The timetable shown above has 4 breaks, each of them marked with a *. Since there are 6 teams, it is not possible to find for this tournament a schedule with less than 4 breaks. □

Assume now that the number of teams is odd. The minimum number of rounds in a single round robin tournament is now larger than $n-1$. If the number of teams is odd, then one team has to remain idle in each round. When a team does not play in one round, it is referred to as a *Bye* (*B*). So when the number of teams is odd, the sequence of games that have to be played by a given team is a string of *H*'s, *A*'s, and one or more *B*'s, e.g., *HAHABA*. With these more complicated types of patterns a break can be defined in several ways. An *HBH* substring may or may not be considered a break; if the *B* is considered equivalent to an *A*, then there is no break. If the *B* is considered equivalent to an *H*, then there is a break (actually, then there are two breaks). However, one can argue that an *HBH* pattern is less bad than an *HHH* pattern; one can even argue that it is less bad than an *HH* pattern. So, as far as penalties or costs are concerned, the cost of an *HBH* pattern may actually be less than the cost of a single break.

It turns out that a single round robin tournament problem with arbitrary n (n being either even or odd) can be described as a graph coloring problem. This equivalence is somewhat similar to the relationships between timetabling problems and graph coloring problems described in a previous section; it provides some additional insight into the tournament scheduling problem as well. Consider a single round robin tournament in which each club has to face every other club once and only once; the game is either a home game (*H*) or an away game (*A*). A directed graph G=(N, B) can be constructed in which set N consists of n nodes and each node corresponds to one team. Each node is linked via an arc to each other node. The arcs are initially undirected. In Figure 2 the n nodes are positioned in such a way that they form a polygon. If in a graph each node is connected to every other node, it is referred to as a clique or as a complete graph.

A well-known graph coloring problem concerns the coloring of the arcs in a graph; the coloring has to be done in such a way that all the arcs that are linked to any given node have different colors and the total number of colors is minimized. It is a well-known fact that a clique with n nodes can be colored this way with n colors (which is often referred to as its *chromatic number*). Each subgraph that receives a specific color consists of one arc that lies on the boundary of the polygon and a number of internal arcs (see Figure 2).

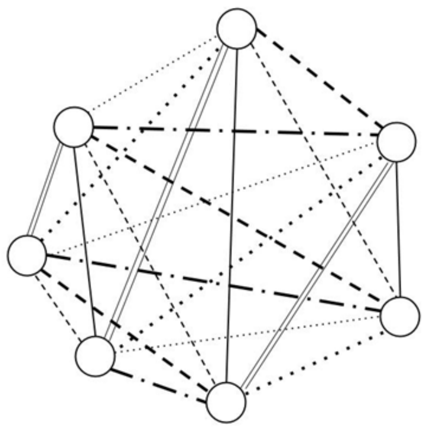

Figure 2. Coloring of a complete graph

The equivalence between the graph coloring problem and the single round robin tournament scheduling problem is based on the fact that each round in the tournament corresponds to a subgraph with a different color. The coloring of the arcs for the different rounds thus determines a schedule for a single round robin tournament in which each team plays every other team only once. One question is how to partition the arcs into a number of subsets with each subset having a different color. A second question has to be addressed as well: when one team plays another it has to be decided at which one of the two sites the game is played, i.e., which team plays at home and which team will be away. In order to determine this, each arc in the graph has to be directed; if a game between teams i and j takes place at team j's site, then the arc linking nodes i and j emanates from i and goes to j. In order to avoid breaks in two consecutive rounds of a schedule, the arcs have to be directed in such a way that the two subgraphs corresponding to the two consecutive rounds in the timetable constitute a so-called *directed Hamiltonian path*. A directed Hamiltonian path is a path that goes from one node to another with each node having at most one outgoing arc and at most one incoming arc.

Many approaches for developing tournament schedules are based on a standard framework for the search for good feasible schedules. In this framework a pattern is equivalent to a string consisting of H's, A's, and B's, for example *HABAHHA*. For a single round robin tournament the length of a string is $n-1$ (n) when the number of teams is even (odd). These strings are often referred to as Home Away Patterns (HAPs). The following three step algorithm provides a framework for generating single round robin schedules.

Step 1 – Assemble a Collection of HAPs

Find a collection of n different HAPs.

This set of HAPs is referred to as the *pattern set*.

Step 2 – Create a Timetable

Assign a game to each entry in the pattern set.

The resulting assignment is referred to as a timetable.

Step 3 – Assign Teams to Patterns

Assign a team to each pattern.

Together with the timetable, this creates a single round robin schedule.

A schedule for a double round robin tournament can be created in a similar fashion. First, a single round robin tournament schedule is generated. Then a fourth step is added, which is typically referred to as the *mirroring step*. The single round robin schedule is extended by attaching immediately behind it a schedule that is exactly the same but with the home and away games reversed.

In practice, the framework described above is often used somewhat differently. In Step 1 usually more than n different HAPs are generated and based on this larger collection of HAPs more than one pattern set is created. Additional pattern sets give more choices and flexibility in the creation of timetables and schedules.

If the tournament under consideration has a large number of teams, then each step in the framework requires a certain computational effort. There are actually various approaches that can be used in each step. Each step can be implemented following either an optimization approach or a constraint programming approach. The remaining part of this section describes the use of optimization techniques in each step of the framework.

In Step 1 several pattern sets can be generated by first listing all the preferred patterns (with alternating H's and A's and one B) of appropriate length. There may not be that many of such preferred patterns. A list of some of the less preferred patterns (say, with one or two breaks) is created as well. It is not likely that a set that consists only of preferred patterns will ultimately lead to an acceptable schedule. Because of this, additional pattern sets are created that contain, for example, n–2 preferred patterns and two patterns that are less preferred. If we allow only a small number of less preferred patterns in a pattern set, then the number of pattern sets that can be generated is still relatively small.

Step 2 creates timetables for different teams. Determining the timetables can also be done through integer programming. It is clear that every pattern in each one of its rounds is linked to another pattern. Let S denote a set of n patterns and let T denote the set of rounds. The binary variable $x_{k\ell t}$ is 1 if the team associated with pattern k plays at the site of the team associated with pattern ℓ in round t. Of course, this variable is only defined if the kth pattern has an A in position t and the ℓth pattern has an H in position t. Let F denote the set of all feasible (k,ℓ,t) triplets. In order to find for a single round robin tournament a solution that satisfies all the constraints the following integer program can be formulated.

$$\text{Minimize} \quad \sum_{(k,\ell,t)\in F} x_{k\ell t}$$

subject to

$$\sum_{t:(k,\ell,t)\in F} x_{k\ell t} + \sum_{t:(\ell,k,t)\in F} x_{\ell kt} = 1, \text{ for all } \; k\in S, \ell \in S, k \neq \ell$$

$$\sum_{\ell:(k,\ell,t)\in F} x_{k\ell t} + \sum_{\ell:(\ell,k,t)\in F} x_{\ell kt} \leq 1, \text{ for all } \; k\in S, t\in T$$

$$x_{k\ell t} \in \{0,1\}, \text{ for all } \; (k,\ell,t)\in F$$

The first set of constraints specifies that during the tournament there will be exactly one game between teams represented by patterns k and ℓ. The second set of constraints specifies that pattern k plays at most one game in round t. (In a single round robin with an even number of teams this inequality constraint becomes an equality constraint.) The objective function for this integer program is somewhat arbitrary, because the only goal is to find a solution that satisfies all constraints.

Step 3 assigns teams to patterns. This step may in certain situations also be formulated as an integer program. Let y_{ik} denote a $0-1$ variable taking value 1 if team i is assigned to HAP k and 0 otherwise. Let c_{ik} denote the relative cost of such an assignment (this relative cost is estimated by taking all stated preferences into account). A timetable for the competition can be constructed as follows.

$$\text{Minimize} \quad \sum_{i=1}^{n}\sum_{k=1}^{n} c_{ik} y_{ik}$$

subject to

$$\sum_{i=1}^{n} y_{ik} = 1 \quad \text{for} \quad k = 1,\ldots,n$$

$$\sum_{k=1}^{n} y_{ik} = 1 \quad \text{for} \quad i = 1,\ldots,n$$

Of course, each team is assigned to one HAP and each HAP is assigned to one team. In practice, the mathematical program formulated for Step 3 is often more complicated. It is usually of a form that is referred to as a Quadratic Assignment Problem.

This approach, in which each step is based on an integer programming technique, has been used in a case discussed by Nemhauser and Trick (1998): The Atlantic Coast Conference (ACC) is a group of nine universities in the southeastern United States that compete against each other in a number of sports. From a revenue point of view, the most important sport is basketball. Most of the revenues come from television networks that broadcast the games and from gate receipts. The tournament schedule has an impact on the revenue stream. Television networks need a regular stream of quality games and spectators want neither too few nor too many home games in any period.

There are numerous restrictions in the form of pattern constraints, game count constraints, and team pairing constraints. The patterns of Home games and Away games is important because of wear and tear on the teams, issues of missing class time, and spectator preferences. No team should play more than two Away games consecutively, nor more than two Home games consecutively. A Bye is usually regarded as an Away game. Similar rules apply to weekend slots (no more than two at Home in consecutive weekends).

This type of scheduling problem can also be solved using a constraint programming approach. As far as Steps 1 and 2 are concerned, the computational effort needed using a constraint programming approach seems to be comparable to the computational effort needed using an integer programming approach. However, as far as Step 3 is concerned the constraint programming technique seems to have a clear edge. Since the integer programming approach in Step 3 is basically equivalent to complete enumeration, it is not surprising that constraint programming can do better, see Schaerf (1999), Aggoun and Vazacopoulos (2004), Henz et al. (2004).

In addition to mathematical programming and constraint programming techniques, many researchers have experimented with heuristic techniques as well, see Anagnostopoulos (2003), Hamiez and Hao (2001), Schonberger et al. (2004).

A significant amount of academic research has also focused on finding good solutions for tournaments of specific types, namely in soccer leagues as well as in other leagues; see, for example, Schreuder (1992), Henz (2001), Bartsch et al. (2004).

6.2 Network Broadcast Scheduling

Scheduling of network television programs, even though different in various respects from interval scheduling and tournament scheduling, does exhibit a number of similarities with both interval scheduling and tournament scheduling. The scheduling horizon is typically one week and the week consists of a fixed number of time slots. A number of shows are available for broadcasting and these shows have to be assigned to the different time slots in such a way that a certain objective function is optimized. Moreover, the assignment of shows to slots is subject to a variety of conditions and constraints. For example, assigning a show to one slot may affect the contribution to the objective function of another show in a different slot. The integer programming formulations are somewhat similar to the integer programming formulations for interval scheduling and tournament scheduling.

Major television networks typically have a number of shows available for broadcasting. Some of these belong to series of half hour shows, while others belong to series of one hour shows. There are shows of other lengths as well. There are a fixed number of 30 minute time slots, implying that some shows require one time slot, while others need two consecutive time slots. If a particular show is assigned to a given time slot, then a certain rating can be expected. The forecasted ratings may be based on past experience with the show and/or the time slot; it may be based on lead-in effects due to the shows immediately preceding it, and it may also be based on shows that competing networks assign to that same slot. The profits of the network depend very much on the ratings and one of the main objectives of the network is to maximize its average ratings.

If the length of program *j* is exactly half an hour, then the binary decision variable x_{jt} is 1 if program j is assigned to slot *t*; if the length of program *j* is longer than half an hour, then the decision variable x_{ji} is 1 if the first half hour of program *j* is assigned to slot t (i.e., broadcasting program *j* may require both slots t and *t*+1, but only the decision variable associated with slot *t* is 1 while the one associated with slot *t*+1 remains zero). Let π_{jt} denote the total profit (or the total ratings) obtained by assigning program *j* to time slot *t*. If program *j* occupies more than one slot, then π_{jt} denotes the profit generated over all slots the program covers. Let *A* denote the set of all feasible assignments (*j, t*). Let the binary variable b_{jtv} be 1 if time slot *v* is filled by program *j* or by part of program *j* because of the assignment (*j, t*) and 0 otherwise. Clearly, $b_{jtt} = 1$ and b_{jtv} can only be nonzero for $v > t$. The following integer program can be formulated to maximize the total profit.

$$\text{Maximize} \quad \sum_{(j,t)\in A} \pi_{jt} x_{jt}$$

subject to

$$\sum_{t:(j,t)\in A} x_{jt} \le 1 \qquad \text{for} \quad j = 1,\dots,n$$

$$\sum_{(j,t)\in A} x_{jt} b_{jtv} = 1 \qquad \text{for} \quad v = 1,\dots,H$$

$$x_{jt} \in \{0,1\} \qquad \text{for} \quad (j,t) \in A$$

This integer program takes into account the fact that there are shows of different durations. However, the formulation above is still too simple to be of any practical use. One important issue in television broadcasting revolves around so-called lead-in effects. These effects may have a considerable impact on the ratings (and the profits) of the shows. If a very popular show is followed by a new show for which it would be hard to forecast the ratings, then the high ratings of the popular show may have a spill-over effect on the new show; the ratings of the new show may be enhanced by the ratings of the popular show. Incorporating lead-in effects in the formulation described above can be done in several ways. One way can be described as follows: let (j,t,k,u) refer to a lead-in condition that involves show *j* starting in slot *t* and show *k* starting in slot *u*. Let $\mathcal{L}$ denote the set of all possible lead-in conditions. The binary decision variable y_{jtku} is 1 if in a schedule the lead-in condition (j,t,k,u) is indeed in effect and 0 otherwise. Let π'_{jtku} denote the additional contribution to the objective function if the lead-in condition is satisfied. The objective function in the formulation above has to be expanded with the term

$$\sum_{(j,t,k,u)\in\mathcal{L}} \pi'_{jtku} y_{jtku}$$

and the following constraints have to be added:

$$y_{jtku} - x_{jt} \le 0 \quad \text{for } (j,t,k,u)\in\mathcal{L}$$

$$y_{jtku} - x_{ku} \le 0 \quad \text{for } (j,t,k,u)\in\mathcal{L}$$

$$-y_{jtku} + x_{jt} + x_{ku} \le 1 \quad \text{for } (j,t,k,u)\in\mathcal{L}$$

$$y_{jtku} \in \{0,1\} \quad \text{for } (j,t,k,u)\in\mathcal{L}$$

The first set of constraints ensures that y_{jtku} can never be 1 when x_{jt} is zero. The second set of constraints is similar. The third set of constraints ensures that y_{jtku} never can be 0 when both x_{jt} and x_{ku} are equal to 1.

A fair amount of research has been done over the years focusing on the scheduling of network television programs, see Horen (1980), Reddy, Aronson and Stam (1998), and Hall, Liu, and Sidney (1998). Actually, research attention has also focused on the scheduling of commercials in broadcast networks, see Bollapragada, Bussieck, and Mallik (2004), Bollapragada, Cheng, Phillips, Garbiras, Scholes, Gibbs, and Humphreville (2002), and Bollapragada and Garbiras (2004).

7. Conclusions

It is clear that scheduling plays a very important role in many service industries. As stated earlier, scheduling problems in practice may be either static or dynamic. The static problems are very similar to the so-called off-line scheduling problems studied in the academic literature; the dynamic scheduling problems are often similar to either the online scheduling problems or the stochastic scheduling problems analyzed in the literature. Most of the problems that have received attention in the research literature, and that are discussed in this tutorial, are of the static deterministic type. The main reason is that these problems are somewhat easier to analyze. However, even though these problems are easier to analyze than their dynamic counterparts, they are still, more often than not, NP-Hard.

Static deterministic problems can often be formulated relatively easily as mathematical programs (in particular, as Mixed Integer Prorams or MIPs) and the typical solution techniques most often used for such integer programs are Branch-and-Bound and Cutting Plane methods. Static deterministic scheduling problems can at times also be formulated as constraint programs. If that is the case, they can be solved using standard constraint programming techniques, see Goltz and Matzke (2001). We have also seen that a variety of static deterministic scheduling problems (including timetabling, interval scheduling, reservation problems, and tournament scheduling) are equivalent to graph theoretic node coloring problems, e.g., chromatic number, or arc coloring problems. If an equivalence can be shown between a static deterministic scheduling problem and a graph coloring problem (which usually is strongly NP-Hard), then a host of heuristics can be used that have been developed and analyzed in the graph theory and computer science literature.

Many of the scheduling problems discussed in this tutorial have, in practice, either dynamic or stochastic aspects. One may be able to analyze such a more general problem as a Markov Decision Process (MDP), which one may be able to formulate as a Linear Program (LP). However, the transformation of an MDP into an LP may bring about an explosion in the dimension of the problem. So, even though an LP may be solvable in polynomial time, since the transformation of the MDP into the LP is not polynomial, it is still very hard to solve the MDP. One also may be able to formulate a dynamic scheduling problem as an online scheduling problem. However, such a problem typically does not allow for an exact optimization technique. Research in such problems have led to interesting heuristics and hybrid techniques in practice.

In scheduling applications in service industries, many different types of algorithms are in use, from the very simplistic to the very sophisticated. The more sophisticated techniques include algorithms for Mixed Integer Programming formulations, that are based on Branch-and-Bound and on Branch-and-Cut. These methods usually have as goal to find a very optimum solution. That is, they are applied when the data available for the problem are known with a high degree of certainty and it is worthwhile finding the best possible solution. Clearly, the amount of time required for developing the code as well as the computation time needed when running the code tend to be significant. Especially in the transportation industries these techniques are widely used. In other application areas, the use of heuristics is significantly more common. The heuristics include local search techniques, priority rules, as well as hybrid (decomposition) techniques that combine priority rules with local search.

There is an entire industry dedicated to the development of decision support systems for scheduling applications in services. The development of such a decision support system is typically a major endeavour. The database and user interface requirements are usually quite elaborate. See, for example, in Figure 3 the user interface of a system developed by MultiModal Applied Systems for train scheduling.

Moreover, such decision support systems typically have to allow for seamless interactive optimization. There are numerous software companies that are specializing in the development of software and decision support systems for the service industries. For example, Mimosa Scheduling Software markets a host of scheduling software applications for the academic and education markets. Their systems focus on teacher scheduling, classroom assignments, and so on. The Totalview System of the IEX Corp. is a system used to schedule the shifts of operators in call centers. The MultiRail System of MultiModal Applied Systems is a widely used decision support system for train scheduling. Jeppesen Systems, a unit of Boeing, develops decision support systems for the scheduling of aircraft and for crew scheduling. Galactix Software markets their Team Sports Scheduling System. Besides these more generic systems that are being marketed by the various software companies (systems that still may need a significant amount of customization upon installation), many purely application-specific systems have been developed as well. For example, the Bremen Public Transport Authority developed its own system, referred to as DISSY, in order to schedule its drivers.

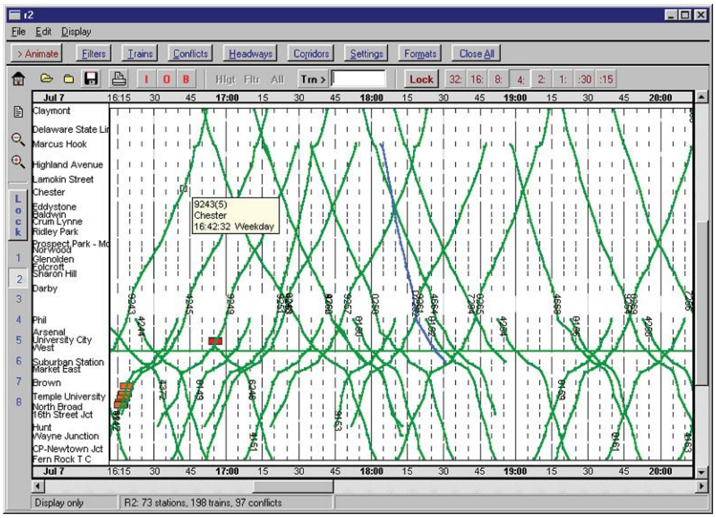

Figure 3. User interface of decision support system for train scheduling

Future research on scheduling in service industries may focus on hybrid and interactive techniques that are useful for systems implementations in practice, e.g., methods that combine constraint programming techniques with optimization techniques or local search techniques (such as genetic algorithms).

Acknowledgments

The work is partially supported by the National Natural Science Foundation of China under Grant No. 71301115, Grant No.71431005.

References

[1] A. Aggoun & A. Vazacopoulos (2004). Solving Sports Scheduling and Timetabling Problems with Constraint Programming. In: S. Butenko, J. Gil-Lafuente & P. Pardalos (eds.), Economics, Management and Optimization in Sports. Springer, New York.

[2] D. Ait-Kadia, J.-B. Menye & H. Kane (2011). Resources assignment model in maintenance activities scheduling. International Journal of Production Research, 49 (22) : 6677–6689.

[3] A. Anagnostopoulos, L. Michel, P. Van Hentenryck & Y. Vergados (2003). A Simulated Annealing Approach to the Traveling Tournament Problem. In Proceedings CPAIOR'03.

[4] C. Archetti, M.G. Speranza & A. Hertz (2006). A tabu search algorithm for the split delivery vehicle routing problem. Transportation Science, 40(1): 64–73.

[5] C. Archetti, M.G. Speranza & M.W.P. Savelsbergh (2008). An optimization-based heuristic for the split delivery vehicle routing problem. Transportation Science, 42(1): 22–31.

[6] P. Augerat, J.M. Belenguer, E. Benavent, A. Corberán, D. Naddef & G. Rinaldi (1998). Computational results with a branch and cut code for the capacitated vehicle routing problem. R. 495, Dicembre 1998, Istituto di Analisi dei Sistemi ed Informatica, CNR, Roma, Italy.

[7] R. Baldacci, A. Mingozzi & R. Roberti (2012). Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. European Journal of Operational Research, 218(1): 1–6.

[8] C. Bandi & D. Bertsimas (2012). Tractable stochastic analysis in high dimensions via robust optimization. Mathematical Programming,134(1): 23–70.

[9] Ph. Baptiste, C.L. Le Pape & W. Nuijten (2001). Constraint-Based Scheduling. Kluwer Academic Publishers, Boston.

[10] C. Barnhart, P. Belobaba & A. Odoni (2003). Applications of Operations Research in the Air Transport Industry. Transportation Science, 37: 368–391.

[11] C. Barnhart & G. Laporte (eds.) (2006). Handbooks in Operations Research and Management Science. Volume 14 – Transportation, North-Holland, Amsterdam.

[12] C. Barnhart, F. Lu & R. Shenoi (1998). Integrated Airline Schedule Planning. In: G. Yu (ed.), Operations Research in the Airline Industry, Chapter 13, pp. 384–403. Kluwer Academic Publishers, Boston.

[13] J.J. Bartholdi III, J.B. Orlin & H.D. Ratliff (1980). Cyclic Scheduling via Integer Programs with Circular Ones. Operations Research, 28: 1074–1085.

[14] T. Bartsch, A. Drexl & S. Kr¨oger (2006). Scheduling the professional soccer leagues of austria and germany. Computers and Operations Research, 33: 1907–1937.

[15] M.A. Begen & M. Queyranne (2011). Appointment scheduling with discrete random durations. Mathematics of Operations Research, 36(2): 240–257.

[16] J. Belien & E. Demeulemeester (2007). Building cyclic master surgery schedules with leveled resulting bed occupancy. European Journal of Operational Research, 176: 1185–1204.

[17] R.E. Bixby & E.K. Lee (1998). Solving a truck dispatching problem using branch-and-cut. Operations Research, 46: 355–367.

[18] J.T. Blake & M.W. Carter (1997). Surgical process scheduling: a structured review. Journal of the Society for Health Systems, 5(3):17–30.

[19] J.T. Blake & M.W. Carter (2002). A goal-programming approach to resource allocation in acute-care hospitals. European Journal of Operational Research, 140: 541–561.

[20] J.T. Blake & J. Donald (2002). Using Integer Programming to Allocate Operating Room Time at Mount Sinai Hospital. Interfaces, 32(2): 63–73.

[21] J. Blazewicz, W. Cellary, R. Slowinski & J. Weglarz (1986). Scheduling under resource constraints - deterministic models. Annals of Operations Research, Vol. 7, Baltzer, Basel.

[22] L. Bodin, B. Golden, A. Assad & M. Ball (1983). Routing and Scheduling of Vehicles and Crews: The State of the Art. Computers and Operations Research, Vol. 10, 63–211.

[23] S. Bollapragada, M. Bussieck & S. Mallik (2004). Scheduling commercial videotapes in broadcast television. Operations Research, 52: 679–689.

[24] S. Bollapragada, H. Cheng, M. Phillips, M. Garbiras, M. Scholes, T. Gibbs & M. Humphreville (2002). NBC's Optimization

Systems Increase Revenues and Productivity. Interfaces, 32(1): 47–60.

[25] S. Bollapragada & M. Garbiras (2004). Scheduling commercials on broadcast television. Operations Research, 52: 337–345.

[26] K.I. Bouzina & H. Emmons (1996). Interval scheduling on identical machines. Journal of Global Optimization, 9: 379–393.

[27] M.L. Brandeau, F. Sainfort & W.P. Pierskalla (eds.) (2004). Operations Research and Health Care – A Handbook of Methods and Applications, Springer.

[28] D. Brelaz (1979). New Methods to Color the Vertices of a Graph. Communications of the ACM, 22: 251–256.

[29] G.G. Brown, G.W. Graves & D. Ronen (1987). Scheduling ocean transportation of crude oil. Management Science, 33: 335–346.

[30] P. Brucker, A. Drexl, R. Mõhring, K. Neumann & E. Pesch (1999). Resource constrained project scheduling: notation, classification, models, and methods. European Journal of Operational Research, 112: 3–41.

[31] W.J. Burgess & R.E. Busby (1992). Personnel Scheduling. In: G. Salvendy (ed.), Handbook of Industrial Engineering chapter 81, pp. 2155–2169. J. Wiley, New York.

[32] E. Burke & M.W. Carter (eds.) (1998). The Practice and Theory of Automated Timetabling II. Selected Papers from the 2nd International Conference on the Practice and Theory of Automated Timetabling (held August 1997 in Toronto), Lecture Notes in Computer Science No. 1408, Springer Verlag, Berlin.

[33] E. Burke & P. De Causmaecker (eds.) (2003). The Practice and Theory of Automated Timetabling IV. Selected Papers from the 4th International Conference on the Practice and Theory of Automated Timetabling, Gent, Belgium, August 2002. Lecture Notes in Computer Science No. 2749, Springer Verlag, Berlin.

[34] E. Burke, P. De Causmaecker, G. Vanden Berghe & H. Van Landeghem (2004). The state of the art of nurse rostering. Journal of Scheduling, 7: 441–499.

[35] E. Burke, D.G. Elliman, P.H. Ford & R.F. Weare (1996). Exam Timetabling in British Universities: A Survey. The Practice and Theory of Automated Timetabling, E. Burke and P. Ross (eds.), Lecture Notes in Computer Science,Vol.1153, Springer Verlag, Berlin.

[36] E. Burke & W. Erben (eds.) (2001). The Practice and Theory of Automated Timetabling III. Selected Papers from the 3rd International Conference on the Practice and Theory of Automated Timetabling (held August 2000 in Konstanz, Germany), Lecture Notes in Computer Science, Vol. 2079, Springer Verlag, Berlin.

[37] E. Burke & P. Ross (eds.) (1996). The Practice and Theory of Automated Timetabling. Selected Papers from the 1st International Conference on the Practice and Theory of Automated Timetabling, Edinburgh, August 1995. Lecture Notes in Computer Science, Vol. 1153, Springer Verlag, Berlin.

[38] E. Burke & H. Rudova (eds.) (2006). The Practice and Theory of Automated Timetabling VI. Selected Papers from the 6th

International Conference on the Practice and Theory of Automated Timetabling, Brno, Czech Republic, August 2006. Lecture Notes in Computer Science, Vol. 3867, Springer.

[39] E. Burke & M. Trick (eds.) (2004). The Practice and Theory of Automated Timetabling V. Selected Papers from the 5th International Conference on the Practice and Theory of Automated Timetabling, Pittsburgh, U.S.A. August 2004. Lecture Notes in Computer Science, Vol. 3616, Springer Verlag, Berlin.

[40] R.N. Burns & M.W. Carter (1985). Work force size and single shift schedules with variable demands. Management Science, 31: 599–608.

[41] R.N. Burns & G.J. Koop (1987). A modular approach to optimal multiple shift manpower scheduling. Operations Research, 35: 100–110.

[42] S. Butenko, J. Gil-Lafuente & P. Pardalos (2004). Economics, Management and Optimization In Sports. Springer Verlag, Berlin.

[43] A.M. Campbell, D. Vandenbussche & W. Hermann (2008). Routing for relief efforts. Transportation Science, 42(2): 127–145.

[44] A. Caprara, M. Fischetti & P. Toth (2002). Modeling and solving the train timetabling problem. Operations Research, 50: 851–861.

[45] M. Carey & D. Lockwood (1995). A model, algorithms and strategy for train pathing. Journal of the Operational Research Society, 46: 988–1005.

[46] M.W. Carter (1986). A survey of practical applications of examination timetabling algorithms. Operations Research, 34: 193–202.

[47] M.W. Carter & C.A. Tovey (1992). When is the classroom assignment problem hard? Operations Research, 40: S28–S39.

[48] A.M. Caunhye, X. Nie & S. Pokharel (2012). Optimization models in emergency logistics: a literature review. Socio-Economic Planning Sciences, 46(1): 4–13.

[49] T. Cayirli & E. Veral (2003). Outpatient scheduling in health care: a review of literature. Production and Operations Management, 12(4): 519–549.

[50] T. Cayirli, K. Khiong Yang & S.A. Quek (2012). A universal appointment rule in the presence of no-shows and walk-ins. Production and Operations Management, 21(4): 682–697.

[51] A. Ceder, B. Golany & O. Tal (2001). Creating bus timetables with maximal synchronization. Transportation Research Part A: Policy and Practice, 35(10): 913–928.

[52] M. Christiansen (1999). Decomposition of a combined inventory and time constrained ship routing problem. Transportation Science, 33: 3–16.

[53] M. Christiansen, K. Fagerholt & D. Ronen (2004). Ship routing and scheduling – status and perspective. Transportation Science, 38: 1–18.

[54] M. Christiansen, K. Fagerholt, B. Nygreen & D. Ronen (2007). Maritime Transportation. In: C. Barnhart and G. Laporte (eds.), Handbooks in Operations Research and Management Science, Volume 14 – Transportation, pp.189–284, North-Holland, Amsterdam.

[55] J.-F. Cordeau, G. Stojkovic, F. Soumis & J. Desrosiers (2001). Benders Decomposition for Simultaneous Aircraft Routing and Crew

Scheduling. Transportation Science, 35: 375–388.

[56] F.H. Cullen, J.J. Jarvis & H.D. Ratliff (1981). Set partitioning based heuristics for interactive routing. Networks, 11: 125–144.

[57] J.R. Daduna, I. Branco & J.M. Pinto Paixao (1995). Computer Aided Scheduling of Public Transport. Lecture Notes in Economics and Mathematical Systems No. 430, Springer Verlag, Berlin.

[58] E.L. Demeulemeester & W.S. Herroelen (2002). Project Scheduling: A Research Handbook, Kluwer Academic Publishers, Boston, MA.

[59] B. Denton, J. Viapiano & A. Vogl (2007). Optimization of surgery sequencing and scheduling decisions under uncertainty. Health Care Management Science, 10:13–24.

[60] G. Desaulniers, J. Desrosiers, Y. Dumas, M. M. Solomon & F. Soumis (1997). Daily aircraft routing and scheduling. Management Science, 43: 841–855.

[61] M. Desrochers & J.-M. Rousseau (1992). Computer Aided Scheduling of Public Transport. Lecture Notes in Economics and Mathematical Systems No. 386, Springer Verlag, Berlin.

[62] D. de Werra (1988). Some models of graphs for scheduling sports competitions. Discrete Applied Mathematics, 21: 47–65.

[63] S.E. Elmaghraby (1967). On the expected duration of PERT type networks. Management Science, 13: 299–306.

[64] H. Emmons (1985). Workforce scheduling with cyclic requirements and constraints on days off, weekends off, and work stretch. IIE Transactions, 17: 8–16.

[65] H. Emmons & R.N. Burns (1991). Off-day scheduling with hierarchical worker categories. Operations Research, 39: 484–495.

[66] M. Fischetti, A. Lodi, S. Martello & P. Toth (2001). A polyhedral approach to simplified crew scheduling and vehicle scheduling problems. Management Science, 47(6): 833–850.

[67] M.L. Fisher & M.B. Rosenwein (1989). An interactive optimization system for bulk-cargo ship scheduling. Naval Research Logistics, 36: 27–42.

[68] P.M. Francis, K.R. Smilowitz & M. Tzur (2008). The period vehicle routing problem and its extensions. The vehicle routing problem: latest advances and new challenges, 73–102, Springer, US.

[69] R. Freling, A.P.M. Wagelmans & J.M.P. Paixao (2001). Models and algorithms for single-depot vehicle scheduling. Transportation Science, 35(2): 165–180.

[70] D.R. Fulkerson (1962). Expected critical path lengths in PERT type networks. Operations Research, 10: 808–817.

[71] M.R. Garey, D.S. Johnson, B.B. Simons & R.E. Tarjan (1981). Scheduling unit-time tasks with arbitrary release times and deadlines. SIAM Journal of Computing, 10: 256–269.

[72] M. Gawande (1996). Workforce Scheduling Problems with Side Constraints. Paper presented at the semiannual INFORMS meeting in Washington DC , May 1996.

[73] H.J. Goltz & D. Matzke (2001). ConBaTT – Constraint Based TimeTabling. In: E. Burke and W. Erben (eds.), The Practice and Theory of Automated Timetabling III. Selected

Papers from the 3rd International Conference on the Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science, Vol. 2079, Springer Verlag, Berlin.

[74] M. Grönkvist (2002). Using Constraint Propagation to Accelerate Column Generation in Aircraft Scheduling. Technical Report 2002, Computing Science Department, Chalmers University of Technology, Gothenburg, Sweden.

[75] D. Gupta & B. Denton (2008). Appointment scheduling in health care: challenges and opportunities. IIE Transactions, 40(9): 800–819.

[76] D. Gupta & W.Y. Wang (2012). Patient appointments in ambulatory care. In: Handbook of Healthcare System Scheduling, 168: 65–104. Springer, New York,.

[77] K. Haase, G. Desaulniers & J. Desrosiers (2001). Simultaneous vehicle and crew scheduling in urban mass transit systems. Transportation Science, 35(3): 286–303.

[78] K. Hãgele, C.O. Dúnlaing & S. Riis (2001). The complexity of scheduling TV commercials. Electronic Notes in Theoretical Computer Science, Vol. 40.

[79] N.G. Hall, W.-P. Liu & J.B. Sidney (1998). Scheduling in broadcast networks. Networks, 32: 233–253.

[80] R. Hall (2012). Handbook of Healthcare System Scheduling, Vol. 168, Springer, New York.

[81] J.-P. Hamiez & J.-K. Hao (2001). Solving the Sports League Scheduling Problem with Tabu Search. In: A. Nareyek (ed.), Local Search for Planning and Scheduling, Lecture Notes in Computer Science, 2148: 24–36. Springer Verlag, Berlin.

[82] E. Hans, G. Wullink, M. van Houdenhoven & G. Kazemier (2008). Robust surgery loading. European Journal of Operational Research, 185: 1038–1050.

[83] R. Hassin & S. Mendel (2008). Scheduling arrivals to queues: a single-server model with no-shows. Management Science, 54(3): 565–572.

[84] M. Henz (2001). Scheduling a major college basketball conference-revisited. Operations Research, 49: 163–168.

[85] M. Henz, T. Müller & S. Thiel (2004). Global constraints for round robin tournament scheduling. European Journal of Operational Research, 153: 92–101.

[86] J. Holguín-Veras, M. Jaller, L.N.V. Wassenhove, N. Pérez & T. Wachtendorf (2012). On the unique features of post-disaster humanitarian logistics. Journal of Operations Management, 30 (7): 494–506.

[87] J. Holguín-Veras, N. Pérez, M. Jaller, L.N.V. Wassenhove & F. Aros-Vera (2013). On the appropriate objective function for postdisaster humanitarian logistics models. Journal of Operations Management, 31(5): 262–280.

[88] J.H. Horen (1980). Scheduling of network television programs. Management Science, 26: 354–370.

[89] V.N. Hsu, R. de Matta & C.-Y. Lee (2003). Scheduling patients in an ambulatory surgical center. Naval Research Logistics, 50: 219–238.

[90] M. Huang, K. Smilowitz & B. Balcik (2012). Models for relief routing: equity, efficiency and efficacy. Transportation research part E: logistics and transportation review , 48 (1): 2–18.

[91] D. Huisman, R. Freling & A.P.M. Wagelmans (2005). Multiple depot integrated vehicle and crew scheduling. Transportation Science,39 (4): 491–502.

[92] R. Hung & H. Emmons (1993). Multiple-shift workforce scheduling under the 3-4 compressed workweek with a hierarchical workforce. IIE Transactions, 25: 82–89.

[93] H.Z. Jia, F. Ordonez & M. Dessouky (2007). A modeling framework for facility location of medical services for large-scale emergencies. IIE transactions, 39 (1): 41–55.

[94] M.Z. Jin, K. Liu & R.O. Bowden (2007). A two-stage algorithm with valid inequalities for the split delivery vehicle routing problem. International Journal of Production Economics, 105 (1): 228–242.

[95] G.C. Kaandorp & G. Koole (2007). Optimal outpatient appointment scheduling. Health Care Management Science, 10 (3): 217–229.

[96] P. Keskinocak & S. Tayur (1998). Scheduling of time-shared jet aircraft. Transportation Science, 32: 277–294.

[97] K.J. Klassen & R. Yoogalingam (2009). Improving performance in outpatient appointment services with a simulation optimization approach. Production and Operations Management, 18 (4): 447–458.

[98] R. Kolish (1995). Project Scheduling under Resource Constraints. Physica-Verlag (Springer Verlag), Heidelberg.

[99] L.R. LaGanga & S.R. Lawrence (2012). Appointment overbooking in health care clinics to improve patient service and clinic performance. Production and Operations Management, 21 (5): 874–888.

[100] G. Laporte & S. Desroches (1984). Examination timetabling by computer. Computers and Operations Research, 11: 351–360.

[101] C.G. Lee, M.A. Epelman, C.C. White III & Y.A. Bozer (2006). A shortest path approach to the multiple-vehicle routing problem with split pick-ups. Transportation research part B: Methodological, 40 (4): 265–284.

[102] C.-Y. Lee & L. Lei (2001). Multiple-project scheduling with controllable project duration and hard resource constraint: Some solvable cases. Annals of Operations Research, 102 (1-4): 287–307.

[103] R.E. Marsten & F. Shepardson (1981). Exact solution of crew scheduling problems using the set partitioning model: recent successful applications. Networks, 11: 165–177.

[104] C.U. Martel (1982a). Scheduling Uniform Machines with Release Times, Deadlines and Due times. In: M.A. Dempster, J.K. Lenstra and A.H.G. Rinnooy Kan (Eds.), Deterministic and Stochastic Scheduling, pp.89–99. Reidel, Dordrecht.

[105] C.U. Martel (1982b). Preemptive scheduling with release times, deadlines and due times. Journal of the Association of Computing Machinery, 29: 812–829.

[106] C. Martin, D. Jones & P. Keskinocak (2003). Optimizing on-demand aircraft schedules for fractional aircraft operators. Interfaces, 33 (5): 22–35.

[107] M. Mesquita & J. Paixao (1999). Exact algorithms for the multi-depot vehicle scheduling problem based on multicommodity network flow type formulations. Computer-aided transit

scheduling, 221–243. Springer Berlin Heidelberg.

[108] W.P. Millhiser & E. Veral (2014). Designing appointment system templates with operational performance targets. Working Paper, Zicklin School of Business, Baruch College CUNY, New York, NY.

[109] R. Miyashiro & T. Matsui (2003). Round-Robin Tournaments with a Small Number of Breaks. Department of Mathematical Informatics Technical Report METR 2003-29, Graduate School of Information Science and Technology, the University of Tokyo, Tokyo, Japan.

[110] J.G. Moder & C.R. Philips (1970). Project Management with CPM and PERT. Van Nostrand Reinhold, New York.

[111] J.M. Mulvey (1982). A classroom/time assignment model. European Journal of Operational Research, 9: 64–70.

[112] R. Nanda & J. Browne (1992). Introduction to Employee Scheduling. Van Nostrand Reinhold, NY.

[113] G.L. Nemhauser & M. Trick (1998). Scheduling a major college basketball conference. Operations Research, 46: 1–8.

[114] K. Neumann, C. Schwindt & J. Zimmermann (2001). Project Scheduling with Time Windows and Scarce Resources. Lecture Notes in Economics and Mathematical Systems No. 508, Springer Verlag, Berlin.

[115] J.H. Patterson (1984). A comparison of exact approaches for solving the multiple constrained resources rroject scheduling problem. Management Science, 30: 854–867.

[116] A.N. Perakis & W.M. Bremer (1992). An operational tanker scheduling optimization system: background, current practice and model formulation. Maritime Policy and Management, 19: 177–187.

[117] PERT (1958). Program Evaluation Research Task. Phase I Summary Report, Special Projects Office, Bureau of Ordnance, Department of the Navy, Washington, D.C.

[118] V.V. Peteghem & M. Vanhoucke (2014). An experimental investigation of metaheuristics for the multi-mode resource-constrained project scheduling problem on new dataset instances. European Journal of Operational Research, 235 (1): 62–72.

[119] M.E. Posner (1985). Minimizing weighted completion times with deadlines. Operations Research, 33: 562–574.

[120] S.K. Reddy, J.E. Aronson & A. Stam (1998). SPOT: scheduling programs optimally for television. Management Science, 44: 83–102.

[121] J.C. Regin (1999). Sports Scheduling and Constraint Programming. Paper presented at INFORMS meeting in Cincinnati, Ohio.

[122] C.C. Ribeiro & F. Soumis (1994). A column generation approach to the multiple-depot vehicle scheduling problem. Operations Research, 42 (1): 41–52.

[123] L.W. Robinson & R.R. Chen (2010). A comparison of traditional and open-access policies for appointment scheduling. Manufacturing & Service Operations Management, 12 (2): 330–346.

[124] L.M. Rousseau, M. Gendreau & G. Pesant (2002). A general approach to the physician rostering problems. Annals of Operations Research, 115: 193–205.

[125] M. Sasieni (1986). A note on PERT times. Management Science, 30: 1652–1653.

[126] A. Schaerf (1999). Scheduling sport tournaments using constraint logic programming. Constraints, 4: 43–65.

[127] J. Schönberger, D.C. Mattfeld & H. Kopfer (2004). Memetic algorithm timetabling for non-commercial sport leagues. European Journal of Operational Research, 153: 102–116.

[128] J.A.M. Schreuder (1992). Combinatorial aspects of construction of competition dutch professional football leagues. Discrete Applied Mathematics, 35: 301–312.

[129] G. Stojkovic, F. Soumis, J. Desrosiers & M. Solomon (2002). An Optimization Model for a Real-Time Flight Scheduling Problem. Transportation Research Part A: Policy and Practice, 36: 779–788.

[130] M. Stojkovic, F. Soumis & J. Desrosiers (1998). The operational airline crew scheduling problem. Transportation Science, 32: 232–245.

[131] F.B. Talbot (1982). Resource constrained project scheduling with time-resource trade-offs: the nonpreemptive case. Management Science, 28: 1197–1210.

[132] J.M. Tien & A. Kamiyama (1982). On manpower scheduling algorithms. SIAM Review, 24: 275–287.

[133] M. Trick (2001). A Schedule-and-Break Approach to Sports Scheduling. In: E.K. Burke and W. Erben (eds.), Practice and Theory of Automated Timetabling III. Selected Papers of Third International Conference PATAT 2000 (Konstanz, Germany), Lecture Notes in Computer Science 2079: 242–253, Springer Verlag, Berlin.

[134] R. Velasquez, R.T. Melo & K.-H. Kuefer (2008). Tactical Operating Theatre Scheduling: Efficient Appointment Assignment. In: J. Kalcsics & S. Nickel (eds.), Operations Research Proceedings 2007. Proceedings of the GOR 2007 Conference (Saarbruecken), pp. 303–310, Springer.

[135] S. Voss & J.R. Daduna (2001). Computer Aided Scheduling of Public Transport. Lecture Notes in Economics and Mathematical Systems No. 505, Springer Verlag, Berlin.

[136] M.R. Walker & J.S. Sayer (1959). Project Planning and Scheduling. Report 6959, E.I. du Pont de Nemours & Co., Inc., Wilmington, Del.

[137] J. Weglarz, J. Jozefowska, M. Mika & G. Walig´ora (2011). Project scheduling with finite or infinite number of activity processing modesA survey. European Journal of Operational Research, 208 (3): 177–205.

[138] J.D. Wiest & F.K. Levy (1977). A Management Guide to PERT/CPM. Prentice-Hall, Englewood Cliffs.

[139] N.H.M. Wilson (1999). Computer Aided Scheduling of Public Transport. Lecture Notes in Economics and Mathematical Systems No. 471, Springer Verlag, Berlin.

[140] C. S. Wong, F. T. S. Chan & S. H. Chung (2013). A joint production scheduling approach considering multiple resources and preventive maintenance tasks. International Journal of Production Research, 51 (3): 883–896.

[141] A. Wren & J.R. Daduna (1988). Computer Aided Scheduling of Public Transport.

Lecture Notes in Economics and Mathematical Systems No. 308, Springer Verlag, Berlin.

[142] G. Yu (ed.) (1998). Operations Research in the Airline Industry. Kluwer Academic Publishers, Boston.

[143] C. Zacharias & M. Pinedo (2014a). Appointment scheduling with no-shows and overbooking. Production and Operations Management, 23 (5): 788–801.

[144] C. Zacharias & M. Pinedo (2014b). Managing customer arrivals in service systems with multiple servers. Working Paper, Stern School of Business, New York University, New York, NY.

Michael Pinedo received the Ir. degree in mechanical engineering from the Delft University of Technology, the Netherlands in 1973 and the M.Sc. and Ph.D. degrees in operations research from the University of California at Berkeley in 1978. He is the Julius Schlesinger Professor of operations management in the Department of Information, Operations and Management Sciences (IOMS) at the Stern School of Business at New York University. His research focuses on the modeling of production and service systems, and, more specifically, on the planning and scheduling of these systems. Recently, his research also has been focusing on operational risk in financial services. He has (co-)authored numerous technical papers on these topics. He is the author of the books "Scheduling: Theory, Algorithms and Systems" and "Planning and Scheduling in Manufacturing and Services". He is Editor of the Journal of Scheduling.

Christos Zacharias is a visiting assistant professor of management science at the University of Miami. He received his PhD in operations management from the Stern School of Business at New York University, and his BSc in Mathematics from the University of Athens, Greece. His research focuses on designing and optimizing service operations, with an emphasis on health care delivery. Broader areas of interest include stochastic modeling, applied probability, scheduling, and simulation.

Ning Zhu is currently an assistant professor at College of Management and Economics, Tianjin University in China. He received the B.S. and M.S. degrees in information systems and information management from the Xi'an University of Technology in 2006 and 2009, respectively, and the Ph.D. degree in management science and engineering from the Tianjin University in 2012. His research interests include modeling and optimization of transportation systems and operations management.